EVALUACIÓN DE SEGURIDAD DE GESTORES DE BASES DE DATOS NOSQL MONGODB, REDIS Y CASSANDRA

DAVOR JULIÁN MORENO BERNAL
RICARDO ANDRÉS GONZÁLEZ SÁNCHEZ

UNIVERSIDAD CATÓLICA DE COLOMBIA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ D.C.
2020

EVALUACIÓN DE SEGURIDAD DE GESTORES DE BASES DE DATOS NOSQL MONGODB, REDIS Y CASSANDRA

DAVOR JULIÁN MORENO BERNAL
RICARDO ANDRÉS GONZÁLEZ SÁNCHEZ

PROYECTO DE GRADO

HECTOR DARIO JAIMES PARADA
Docente Ing. de Sistemas y Computación

GRUPO DE INVESTIGACIÓN TELSAF
KERBEROS

UNIVERSIDAD CATÓLICA DE COLOMBIA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
BOGOTA D.C.
2020

creative commons

Atribución-NoComercial 2.5 Colombia (CC BY-NC 2.5)

La presente obra está bajo una licencia:
Atribución-NoComercial 2.5 Colombia (CC BY-NC 2.5)
Para leer el texto completo de la licencia, visita:
http://creativecommons.org/licenses/by-nc/2.5/co/

Usted es libre de:

Compartir - copiar, distribuir, ejecutar y comunicar públicamente la obra

hacer obras derivadas

Bajo las condiciones siguientes:

Atribución — Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciante (pero no de una manera que sugiera que tiene su apoyo o que apoyan el uso que hace de su obra).

No Comercial — No puede utilizar esta obra para fines comerciales.

Nota de Aceptación

_____

_____

_____

_____

Presidente del Jurado

_____

Jurado

_____

Jurado

Bogotá D.C, 17 de noviembre del 2020

Dedicatoria

Este trabajo va a dedicado a todas esas personas ya sean amigos, compañeros, docentes, familiares, etc. Que dieron su apoyo, su confianza y su compañía a lo largo de estos 5 años de carrera universitaria.

AGRADECIMIENTOS

A mi padre Ricardo González y a mi madre Mireya Sánchez les agradezco a su esfuerzo y dedicación en formación ya que han sido indispensables para culminar mi carrera universitaria y por todo el apoyo que me han dado ya que fue suficiente para no decaer cuando se presentaba algún problema.

A mi madre Verónica Bernal Rojas por ser esa mujer que siempre lo ha dado todo por mí, por enseñarme que incluso la tarea más grande se puede lograr con esfuerzo y dedicación y sobre todo por su gran amor de madre.

Al ingeniero Héctor Darío Jaimes Parada por ser el encargado de orientarnos en todos los momentos que necesitamos, no solo en la elaboración de este trabajo de grado, sino a lo largo de nuestra carrera universitaria y habernos brindado el apoyo para desarrollarnos profesionalmente.

A Dios por la oportunidad de crecer académicamente en un ambiente universitario que honra sus enseñanzas y por pertenecer a una universidad donde se comparte con docentes, administradores y estudiantes el objetivo de ayudar a mejorar la sociedad.

CONTENIDO

Pág.

LISTA DE TABLAS

Pág

LISTA DE GRÁFICAS

Pág

LISTA DE FIGURAS

Pág

RESUMEN

El presente proyecto busca realizar una evaluación de la seguridad de gestores de bases de datos NoSQL. La metodología de trabajo se desarrolla según el orden de los objetivos, inicia sintetizando los tipos de vulnerabilidades, ataques y esquemas de protección limitado a MongoDB, Redis y Apache Cassandra. Una vez establecidos se diseñará un prototipo de un sistema web que almacena información con una base de datos de tipo no relacional sobre la cual se aplicarán una serie de ataques definidos por un plan de pruebas buscando agregar, consultar, modificar o eliminar información. Finalmente se presentará un informe que exponga los ataques realizados, la manera en que aplicaron, los resultados, posibles contramedidas, ventajas y desventajas de seguridad para cada gestor y las conclusiones obtenidas. Es así que se puede seleccionar cuál herramienta es más conveniente utilizar para una persona u organización en un caso particular. Los resultados demostraron que MongoDB es más vulnerable a ataques de inyección NoSQL, Redis es más vulnerable a ataques registrados en el CVE y que Cassandra es más complejo de utilizar pero es menos vulnerable.

El avance de la tecnología en la creación de nuevas herramientas para solucionar problemas como el almacenamiento de información genera que de manera proporcional se desarrollen métodos que busquen fallas o brechas de seguridad que comprometan dicha información. La necesidad de generar informes de seguridad de manera periódica sobre gestores de bases de datos viene dada por la complejidad y cantidad de ataques que se pueden desarrollar actualmente.

Palabras Clave: Bases de datos, privacidad de los datos, seguridad de los datos, sistema de información.

1. INTRODUCCIÓN

Actualmente a nivel global existe una gran cantidad de entidades que influyen en el funcionamiento general de la sociedad, por ejemplo; Los bancos, hospitales, pymes, áreas gubernamentales o instituciones educativas, como es de esperar dichas entidades manejan información, que se compone de datos, que son la base de su funcionamiento. Para entender cómo se genera un impacto en las áreas de seguridad de información digital es necesario explicar cómo funcionan los gestores de bases de datos en la actualidad, por lo que serán descritos a continuación los conceptos de sistemas de gestión de bases de datos relacionales (SQL) y no relacionales (NoSQL).

En su tercera década de existencia, el lenguaje SQL ofrece una gran flexibilidad a los usuarios soportando bases de datos distribuidas, es decir, que se pueden ejecutar en varias redes de ordenadores a la vez. También se ha convertido en un estándar de lenguaje de consulta para base de datos, siendo la base de una gran variedad de aplicaciones.

El uso de sistemas de gestión de bases de datos relacionales tradicionales llevaría a la creación de una nueva propuesta. Las bases de datos NoSQL no aseguran atributos como atomicidad, consistencia, aislamiento y durabilidad a diferencia de su predecesora, sin embargo, se compensa con la capacidad que manejan dichos sistemas de ofrecer crecimiento a escala y el poder manejar enormes cantidades de datos.

El impacto de los estudios de seguridad como el que se ejecuta en el presente proyecto y para este caso, por medio de una investigación tecnológica para reunir información que permita ejecutar un plan de pruebas en seguridad, para después desarrollando un prototipo aplicar una evaluación de seguridad, tiene relevancia de manera directa en el funcionamiento completo de la sociedad a nivel económico. Por ejemplo, las grandes entidades que se encargan de almacenar información financiera de cada persona en el mundo, debido a la cantidad de datos que trabajan, es probable que en cierto punto necesiten una transición completa a sistemas NoSQL y es necesario poder garantizar la confidencialidad de dicha información, de igual manera las redes sociales se han convertido en un pilar fundamental en la comunicación global y como parte de su requerimiento fundamentales es la confidencialidad de la información de dichos usuarios.

2. OBJETIVOS

2.1 Objetivo general

Evaluar la seguridad de los siguientes sistemas gestores de bases de datos NoSQL: MongoDB, Cassandra y Redis por medio de pruebas sobre entornos controlados para formular contra medidas que mitiguen los riesgos de seguridad en este tipo de sistemas.

2.2 Objetivos específicos

- Recopilar los tipos de vulnerabilidades, ataques y esquemas de protección utilizados en bases de datos NoSQL por medio de revisión documental de estándares y bases de datos científicas.
- Diseñar un plan de pruebas a partir del levantamiento de información para validar la seguridad de un prototipo de un sistema de tipo NoSQL.
- Desarrollar un aplicativo para cada una de las bases de datos NoSQL para ejecutar el plan de pruebas planteado.
- Analizar los resultados obtenidos en la ejecución de las pruebas de cada uno de los gestores para proponer contramedidas que mitiguen los riesgos a los que son expuestos estos tipos de sistemas.

3. PLANTEAMIENTO DEL PROBLEMA

3.1 Definición del problema

A nivel global en la actualidad el manejo de la información ha tomado un valor imprescindible debido a la relevancia que representa tanto la obtención, la modificación o la pérdida de datos para cualquier entidad o incluso personas, es así que es necesario evaluar periódicamente la seguridad de los sistemas que administran dichos datos.

Las causas de las fallas de seguridad en organizaciones, en especial las más pequeñas, son que no se conoce de qué manera se almacena la información que manejan, que se realizan monitoreos de seguridad sin planificar o utilizar estándares, que no se realiza un seguimiento de los permisos de los usuarios que ya no hacen parte del sistema e incluso que no se aplican las correcciones en las fallas dentro de un rango oportuno de tiempo. El poco conocimiento en esta área tan compleja se debe a que en el mercado siempre se encontrarán diversas herramientas para el manejo de información y cada herramienta siempre utilizara un enfoque particular, bajo esta premisa y teniendo en cuenta el objetivo del proyecto, una organización de gran impacto o una compañía pequeña debe considerar entre otros elementos que tan seguro puede ser un gestor de bases de datos NoSQL.

Solucionar todas las fallas de seguridad es muy poco probable debido a la cantidad de personas que trabajan en las áreas de protección de seguridad contra las personas que practican el crimen informático. Es necesario implementar simulaciones de ataques sobre entornos controlados de software para poder encontrar y solucionar errores permitiendo reducir el alcance del problema.
(Poggi, 2018)

InfoSecurity magazine reporta que “Durante el último semestre del año 2017 Victor Gevers, presidente de GDI Foundation y abocado a la divulgación responsable de vulnerabilidades, junto a Dylan Katz, han estado alertando en Twitter ataques hacia bases de datos MongoDB, que apuntan a decenas de miles de servidores con tácticas extorsivas. Durante uno de los ataques registrados se comprometió a 22.000 servidores”.
(Pagnotta, 2017)

Se debe resaltar que, aunque existan ataques sobre este tipo de bases de datos no se deben pasar por alto sus ventajas de uso como poder utilizar sistemas distribuidos, su flexibilidad en el manejo de información y su alto rendimiento a la hora de realizar consultas en robustas bases de datos.

El sitio “Improving Security Together” sintetiza que el CVSS “es un marco abierto para comunicar las características y la gravedad de las vulnerabilidades de software. CVSS consta de tres grupos métricos: base, temporal y ambiental. El grupo Base representa las cualidades intrínsecas de una vulnerabilidad que son constantes a lo largo del tiempo y en los entornos de usuario, el grupo Temporal refleja las características de una vulnerabilidad que cambian con el tiempo y el grupo Ambiental representa las características de una vulnerabilidad que son exclusivas del ambiente en el que se encuentra el usuario.” (First, 2020)

Según análisis realizados y ofrecidos en la página de MongoDB para la fecha 9 de noviembre del año 2019 que fueron calificados por medio del sistema de puntaje CVSS se obtuvo como calificación para el CVE-2019-2390 6.8 en una escala de seguridad del 1 al 10 (MongoDB Alerts, 2020):

- El Impacto de la confidencialidad: Parcial (hay una divulgación informativa considerable).
- Impacto de integridad: Parcial (es posible modificar algunos archivos o información del sistema, pero el atacante no tiene control sobre lo que puede modificarse o el alcance de lo que puede afectar el atacante es limitado).
- Impacto de disponibilidad: Parcial (hay un rendimiento reducido o interrupciones en la disponibilidad de recursos).
- Complejidad de acceso: Medio (Las condiciones de acceso son algo especializadas. Algunas condiciones previas deben ser satisfechas para explotar)
- Autenticación: No se requiere (no se requiere autenticación para aprovechar la vulnerabilidad).
- Acceso obtenido: Ninguna.
  (CVE, D., 2390)

Por su parte Cassandra recibe una calificación de 7.5 para el CVE-2018-8016 (CVE, D., 8016):

- Impacto de confidencialidad: Parcial (hay una divulgación informativa considerable).

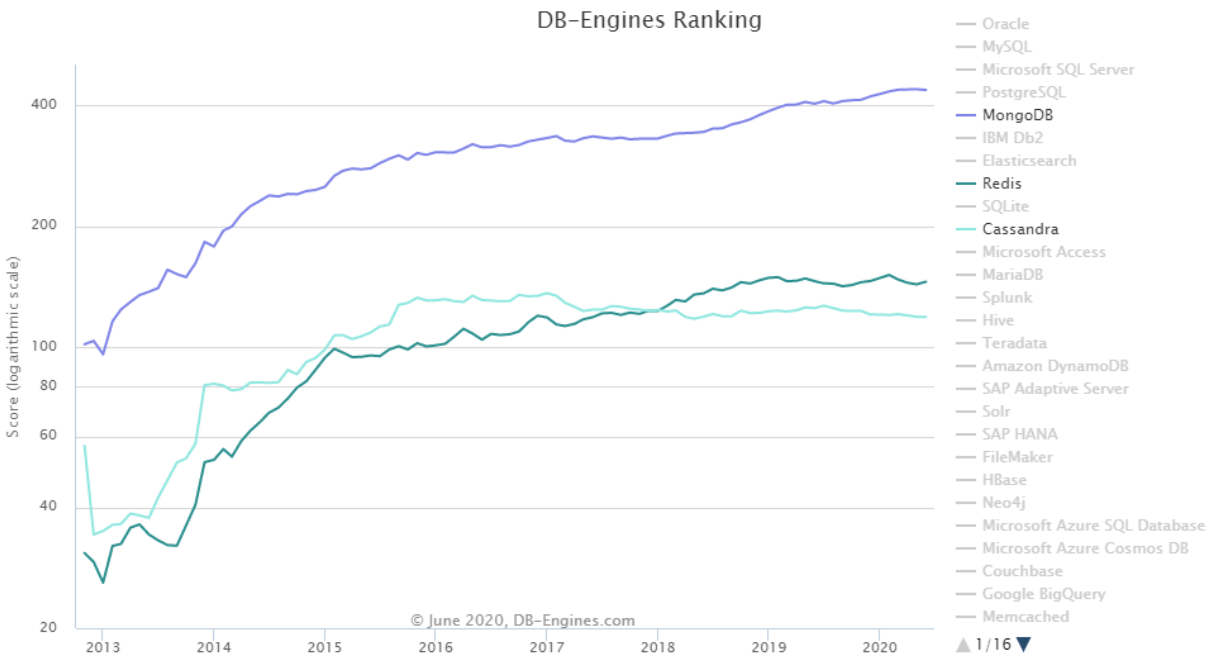
- Impacto de integridad: Parcial (es posible modificar algunos archivos o información del sistema, pero el atacante no tiene control sobre lo que puede modificarse o el alcance de lo que puede afectar el atacante es limitado).
- Impacto de disponibilidad: Parcial (hay un rendimiento reducido o interrupciones en la disponibilidad de recursos).
- Complejidad de acceso: Bajo (no existen condiciones de acceso especializadas o circunstancias atenuantes. Se requiere muy poco conocimiento o habilidad para explotar).
- Autenticación: No se requiere (no se requiere autenticación para aprovechar la vulnerabilidad).
- Acceso obtenido: Ninguna.
- Tipo (s) de vulnerabilidad: Código de ejecución.

Y Redis recibe una calificación de 6.5 para el CVE-2019-10193 (CVE, D., 10193).

- Impacto de confidencialidad: Parcial (hay una divulgación informativa considerable).
- Impacto de integridad: Parcial (es posible modificar algunos archivos o información del sistema, pero el atacante no tiene control sobre lo que puede modificarse o el alcance de lo que puede afectar el atacante es limitado).
- Impacto de disponibilidad: Parcial (hay un rendimiento reducido o interrupciones en la disponibilidad de recursos).
- Complejidad de acceso: Bajo (no existen condiciones de acceso especializadas o circunstancias atenuantes. Se requiere muy poco conocimiento o habilidad para explotar).
- Autenticación: Sistema único (la vulnerabilidad requiere que un atacante inicie sesión en el sistema (como en una línea de comandos o mediante una sesión de escritorio o una interfaz web)).
- Acceso obtenido: Ninguna.
- Tipo (s) de vulnerabilidad: Desbordamiento.

Al analizar los resultados obtenidos se evidencia que para el mes de abril del año 2020 sobre MongoDB se han registrado tres ataques de variados puntajes lo que indica que se trabaja con frecuencia en esta base de datos. Por su parte Apache Cassandra registra para la misma fecha sus dos últimos ataques registrados en el año 2019 y con puntajes críticos lo cual indica que no han hallado fallas en el último año o que no se ha trabajado con frecuencia esta herramienta como para generar reportes de seguridad. Y finalmente Redis registra dos últimos ataques para el año del 2019 de igual manera con puntajes críticos.

Grafica 1 - Ranking BD



(BD-Engines, 2020)

Adicionalmente gracias a la gráfica suministrada por la página DB-Engines.com se puede contemplar el crecimiento en la popularidad de MongoDB, Cassandra y Redis, lo que refleja el uso desde el año 2013 hasta el año 2020 de las bases de datos NoSQL, que se van a trabajar en el proyecto.

De acuerdo a lo presentado se evidencia que MongoDB, Cassandra y Redis, aunque han tenido un aumento de su uso en los últimos años junto a que existen grandes ventajas en el manejo de información también presentan varios problemas correspondientes al área de seguridad. Se recalca el hecho de que son bases de datos que tienen una tendencia importante de uso y por estas razones es necesario hacer estudios de seguridad.

Es de esperar con toda la información presentada que las bases de datos de una organización siempre sean objetivos de intento de robo, debido a información que contiene de las compañías que proveen un servicio y de los clientes que utilizan dichos servicios. La importancia de este hecho se refleja en los usuarios y en la confiabilidad que se tiene de una compañía. De igual manera el aporte de las bases de datos NoSQL en su implementación hacia las pymes es limitado considerando las ventajas que proveen es así que adicionalmente el desarrollo de este tipo de

proyectos representa una ventaja para informar a más personas del uso de estas herramientas.

3.2 Formulación del problema

¿Cómo evaluar la seguridad de los gestores de bases de datos NoSQL MongoDB, Redis y Cassandra por medio de pruebas de intrusión sobre entornos controlados para formular contramedidas?

4. JUSTIFICACIÓN

Como método para controlar el robo de información se han implementado soluciones legales que proponen garantizar sanciones para perpetradores y compensaciones o seguridad para las víctimas. En la ley se encuentran estatutos como la ley 1581 del 2012 enfocados a sentenciar delitos contra la confidencialidad, la integridad y la disponibilidad de los datos. (MinTic, 2012)

Aunque se han tomado medidas para garantizar la seguridad de la información en el aspecto compensatorio o posterior al incidente sucedido es preciso generar soluciones de tipo preventivo que permitan a una persona interactuar por medio de un sistema tecnológico con tranquilidad y confianza. Es en este punto que la seguridad informática toma un papel significativo en la problemática protegiendo la integridad y la privacidad de la información que existe en un sistema tecnológico.

Según una encuesta presentada en 2017 por la página oficial de StackOverflow, se indica que en el quinto lugar y con un 21% de los usuarios utilizan MongoDB para la gestión de información de bases de datos. Si se analiza más profundamente se puede concluir que aún debido a su menor tiempo en el mercado y menor documentación disponible, es un gestor de bases de datos no relacional ocupa la quinta posición de la encuesta esto recalca la relevancia de garantizar seguridad informática para este gestor de base de datos.
(StackOverflow, 2017)

Al usar el paradigma del correcto uso de la información, las empresas pequeñas optan por usar gestores de bases de datos. Por ejemplo, las Pymes recién creadas eligen las nuevas tecnologías NoSQL, o las que han aumentado su escala desean cambiar sus sistemas de almacenamiento realizando transiciones de bases de datos de tipo relacional a tipo no relacional ya que la estructura que maneja al procesamiento de datos es más rápida. Además de estas consideraciones el factor derivado de saber que herramienta puede garantizar mejor la seguridad de la información que otra ocupa un papel fundamental.
Sin embargo, hay dos grandes inconvenientes que hacen que las Bases de Datos NoSQL o no relacionales no avancen a un mayor ritmo, y ambos están relacionados con la formación de las personas que las utilizan.

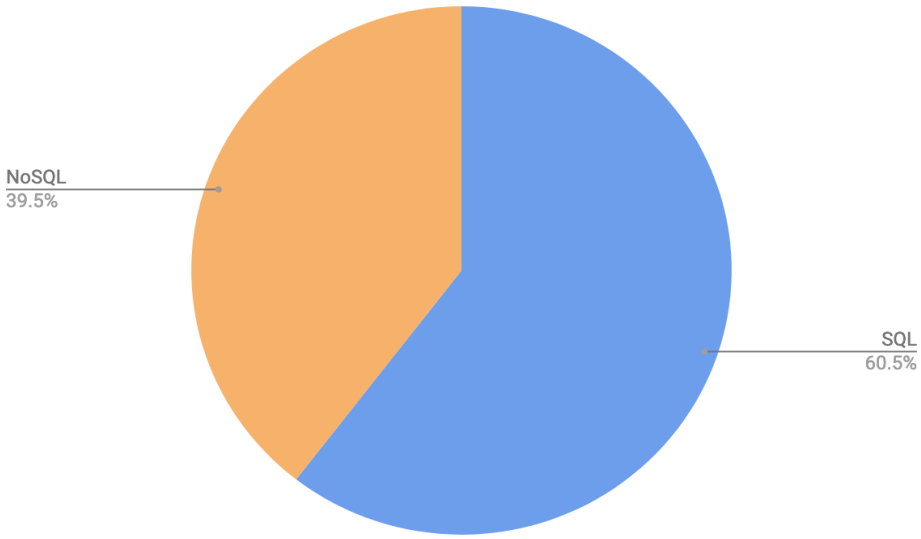
“El primer inconveniente es que muchos expertos se muestran reacios a utilizarlas, en muchas ocasiones, porque no conocen todas las posibilidades que ofrecen; el segundo (relacionado con el primero) es que este tipo de Bases de Datos tienen

una gran diversidad, y hay que tener conocimientos sobre todas ellas para saber cuál es la adecuada para cada caso (es imprescindible tener muy definido el tipo de proyecto al que se asociará, la cantidad esperada de usuarios y la tecnología que se va a utilizar para el desarrollo)".
(DATA, 2015)

Al usar esta herramienta los principales beneficiados según el uso que se le dará a su plataforma son las empresas, por ejemplo, Cassandra es principalmente usada por aplicaciones de redes sociales tales como Facebook y Twitter, por otra parte, MongoDB es utilizado por grandes compañías como Uber. La característica principal del manejo de bases de datos no relacionales es la capacidad de manejar grandes volúmenes de información y su capacidad para expandirse además de que Cassandra se originó como una solución para el manejo de datos de Facebook debido al crecimiento inesperado que tendría la compañía.

ScaleGrid un proveedor de servicios de alojamiento en la nube y de servicios de bases de datos expone en su artículo "2019 Database Trends – SQL vs. NoSQL, Top Databases, Single vs. Multiple Database Use" el uso de las bases de datos comparándolas entre los tipos relacionales y no relacionales. En los resultados presentados se demuestra que si bien el uso de base de datos tradicionales relacionales sigue siendo más popular la diferencia recae en cerca del 10% indicando que el uso de estos sistemas representa una gran cantidad de los usuarios.

Grafica 2 - SQL vs. NoSQL



(ScaleGrid, 2019)

“Hasta la fecha, más de 100,000 bases de datos MongoDB han sido secuestradas. Los hackers también han allanado miles de servidores que alojan datos en ElasticSearch, Apache CouchDB y Hadoop. Los piratas informáticos eliminaron todos los datos de la base de datos y mantuvieron a sus negocios como rehenes al exigir dinero a cambio de la restauración de su base de datos. Si bien la demanda de rescate en sí suele estar en el rango de unos pocos cientos de dólares, tener que revelar una violación de datos puede ser devastador para la reputación de una empresa. No hay garantías de que pagar el rescate recuperará los datos o evitará que se filtren.” (RavenDB, 2020)

Finalmente se destaca el gran y rápido crecimiento a nivel global del uso de las bases de datos NoSQL, su útil variedad de tipos de bases de datos y sus destacables ventajas que pese a su capacidad de optimizar el trabajo no son aprovechadas dado el poco conocimiento de la mayoría de los usuarios. Como se mencionó previamente los problemas de seguridad en las bases de datos NoSQL se deben a varios factores entre los que se encuentran: Desinformación o poca documentación de los gestores, optar por una configuración por defecto, no configurar Firewalls correctamente, no encriptar los datos o no realizar auditorías periódicamente. Con la propuesta actual se busca generar un análisis comparando las fortalezas y debilidades en cuanto a seguridad, con el fin de informar las ventajas o desventajas de cada uno de los gestores en unas áreas específicas. Por ejemplo, de esta manera una compañía puede con base en los resultados obtenidos seleccionar un gestor de bases de datos que solucione ciertas necesidades dentro del área de seguridad que requieran.

5. MARCO REFERENCIAL

5.1 Marco conceptual

- Atributos: Los atributos son las características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades. Ej. Entidad cliente (nombre, apellido, dirección, teléfono). Los atributos se guardan en variables denominadas de instancia, y cada objeto particular puede tener valores distintos para estas variables. (EcuRed, 2020)

- Administrador de Bases de Datos: son responsables del manejo, mantenimiento, desempeño y de la confiabilidad de bases de datos. Asimismo, están a cargo de la mejora y diseño de nuevos modelos de las mismas.
  Manejar una base de datos implica recolectar, clasificar y resguardar la información de manera organizada, por ello, estos profesionales velan por garantizar que la misma esté debidamente almacenada y segura, además de que sea de fácil acceso cuando sea necesario. La mayoría de las empresas alrededor del mundo tienen algún tipo de base de datos digital, por lo que requieren de especialistas en el área para formar parte de su personal. Las bases de datos son comúnmente utilizadas para la gestión de nóminas, registros de clientes, inventarios, etc. (Nuevoo, 2020)

- Aislamiento de privilegios: La gestión de derechos de procesos permite restringir procesos en el nivel de comando, usuario, rol o sistema. Oracle Solaris implementa la gestión de derechos de procesos a través de privilegios. Los privilegios disminuyen el riesgo de seguridad asociado a un usuario o un proceso que tiene capacidades completas de supe usuario en un sistema. Los privilegios y RBAC ofrecen un modelo alternativo eficaz al modelo de supe usuario tradicional. (Oracle, 2020)

- Amenaza: Una Amenaza es la posibilidad de ocurrencia de cualquier tipo de evento o acción que puede producir un daño (material o inmaterial) sobre los elementos de un sistema, en el caso de la Seguridad Informática, los Elementos de Información. Debido a que la Seguridad Informática tiene como propósitos garantizar la confidencialidad, integridad, disponibilidad y autenticidad de los datos e informaciones, las amenazas y los consecuentes daños que puede causar un evento exitoso, también hay que ver en relación

con la confidencialidad, integridad, disponibilidad y autenticidad de los datos e informaciones. (WordPress, 2020)

- AWS (Amazon Web Services): Amazon Web Services (AWS) es la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 175 servicios integrales de centros de datos a nivel global. Millones de clientes, incluyendo las empresas emergentes que crecen más rápido, las compañías más grandes y los organismos gubernamentales líderes, están utilizando AWS para reducir los costos, aumentar su agilidad e innovar de forma más rápida. (AWS, 2020)

- Balanceador de carga: El balanceo de carga (load Balance) se refiere a la distribución del tráfico de red entrante a través de un grupo de servidores backend, también conocido como Server Farm (conjunto de servidor) o Server Pool (conjunto de servidores).
  Los sitios web modernos de alto tráfico deben atender a cientos de miles (algunos hasta millones) de solicitudes concurrentes de usuarios o clientes y devolver los textos, imágenes, videos o datos de aplicaciones correspondientes, todo de manera rápida y confiable. Para lograrlo de forma rentable y cumplir con estos altos volúmenes, la mejor práctica de informática moderna generalmente requiere agregar más servidores.
  Un balanceador de carga actúa como el «Oficial de tránsito» frente a sus servidores y en ruta las solicitudes de los clientes en todos los servidores para satisfacer esas solicitudes de manera que maximice la velocidad y la capacidad para poder garantizar que ningún servidor esté sobrecargado, ya que la saturación podría afectar el rendimiento. (Quanti, 2020)

- Buena práctica de codificación: Además de una correcta y ordenada estructura general que deben tener los programa, es conveniente mantener ciertas buenas prácticas de codificación y el estilo de codificación recomendado. Estas normas no son obligatorias, como lo es la propia sintaxis del lenguaje, pero conviene seguir las recomendaciones de los desarrolladores de Python para facilitar la lectura del programa y ayudar a encontrar posibles errores. Un ejemplo básico para entender a lo que nos referimos es el sangrado, que como hemos visto en Python es obligatorio, pero mientras la estructura de bloques sea correcta, a Python no le importa el número de espacios que se usen. Pues bien, aunque a Python le da igual,

la recomendación es usar cuatro espacios (no tabuladores) para sangrar bloques.
La mayoría de los ataques se realizan debido a una mala desinfección. Un código debidamente validado puede reducir el riesgo de ataques. El uso de una sintaxis bien formada, un formato JSON fuerte, bibliotecas probadas, etc. minimiza el daño en el sistema. (Perez Prieto, 2020)

- Cassandra: es una base de datos no relacional distribuida y basada en un modelo de almacenamiento de “clave - valor” su lenguaje de programación es java y es de libre uso.
  Dentro de los nuevos sistemas de almacenamiento que están surgiendo dentro del universo Big Data, Cassandra es uno de los más interesantes y reseñables. Cassandra se define como una base de datos NoSQL distribuida y masivamente escalable, y esta es su mayor virtud desde nuestro punto de vista, la capacidad de escalar linealmente.
  Además, Cassandra introduce conceptos muy interesantes como el soporte para multi data center o la comunicación peer-to-peer entre sus nodos. (Zaforas, 2016)

- Conciencia: Los programas empresariales de concientización de seguridad de la información son actualmente un elemento fundamental para garantizar la estabilidad en las operaciones de una empresa, por lo que es realmente preocupante que muchas empresas sigan sin brindarles estas herramientas a sus empleados; además, aunque una empresa cuente con cursos de concientización en ciberseguridad, no siempre se encuentran actualizados ni son impartidos de la manera apropiada.
  Especialistas en el diseño de programas de concientización de seguridad informática comentan que la falta de actualización de contenido es el principal problema pues, en la mayoría de las ocasiones, las empresas recurren al uso de material diseñado hace al menos diez años, lo que entorpece la implementación de un adecuado programa de concientización de seguridad de la información. (IICS, 2020)

- Confidencialidad: consiste en la capacidad de garantizar que la información, almacenada en el sistema informático o transmitida por la red, solamente va a estar disponible para aquellas personas autorizadas a acceder a dicha información, es decir, que, si los contenidos cayesen en manos ajenas, estas no podrían acceder a la información o a su interpretación. Este es uno de los principales problemas a los que se enfrentan muchas empresas; en los

últimos años se ha incrementado el robo de los portátiles con la consecuente pérdida de información confidencial, de clientes, líneas de negocio …etc. (INFOSEGUR, 2013)

- Contramedida: La seguridad informática o seguridad de tecnologías de la información es el área de la informática que se enfoca en la protección de la infraestructura computacional y todo lo relacionado con esta y, especialmente, la información contenida o circulante. Para ello existen una serie de estándares, protocolos, métodos, reglas, herramientas y leyes concebidas para minimizar los posibles riesgos a la infraestructura o a la información. La seguridad informática comprende software (bases de datos, metadatos, archivos), hardware y todo lo que la organización valore (activo) y signifique un riesgo si esta información confidencial llega a manos de otras personas, convirtiéndose, por ejemplo, en información privilegiada. (segurinformacion, 2018)

- Desempeño: Uno de los problemas más comunes es el lograr que nuestra base de datos cuente con un desempeño óptimo. Cuando diseñe una base de datos, debe asegurarse de que realiza todas las operaciones importantes de forma rápida y correcta. Algunos problemas de rendimiento se pueden resolver una vez que la base de datos se encuentra en producción. Sin embargo, otros pueden ser el resultado de un diseño inadecuado y se pueden solucionar mediante el cambio de la estructura y el diseño de la base de datos. (TechNet, 2016)

- Disponibilidad: La disponibilidad es una de las características de las arquitecturas empresariales que mide el grado con el que los recursos del sistema están disponibles para su uso por el usuario final a lo largo de un tiempo dado. Ésta no sólo se relaciona con la prevención de caídas del sistema (también llamadas tiempos fuera de línea, downtime u offline), sino incluso con la percepción de “caída” desde el punto de vista del usuario: cualquier circunstancia que nos impida trabajar productivamente con el sistema – desde tiempos de respuesta prolongados, escasa asistencia técnica o falta de estaciones de trabajo disponibles – es considerada como un factor de baja disponibilidad. (everac99, 2008)

- Diseño: El diseño de base de datos es un proceso fundamental a la hora de modelar nuestros conjuntos de datos y definir las operaciones que queremos realizar sobre ellos. Los datos son el activo más importante de nuestra

organización y una base de datos bien diseñada influye de forma directa en la eficiencia que obtendremos a la hora de almacenar, recuperar y analizar nuestros datos. (Carisio, 2020)

- Escaneo de seguridad: es una aplicación diseñada para realizar análisis automáticos de cualquier aplicación, sistema o red en busca de cualquier posible vulnerabilidad. Aunque estas aplicaciones no son capaces de detectar la vulnerabilidad con total precisión, sí son capaces de detectar ciertos elementos que podrían desencadenar en una vulnerabilidad, facilitando enormemente el trabajo a los investigadores e ingenieros. Es frecuente hacer escaneos de vulnerabilidades desde la red interna, para ver que se puede hacer una vez que se tiene acceso a la intranet, o desde red externa, para ver las posibilidades que tiene un atacante externo para atacar nuestros sistemas. (Wiki, 2020)

- Ethical Hacking: Un Hacker Ético es una persona que realiza pruebas de penetración. Es un experto en computadoras y redes de datos cuya función es atacar los sistemas de seguridad en nombre de los dueños con la intención de buscar y encontrar vulnerabilidades actuando de forma igual o al menos similar y utilizan los mismos métodos que usaría un hacker con intención de atacar el sistema. En pocas palabras vulneran y entran a los sistemas empresariales para reportar vulnerabilidades, en lugar de robarla o borrarla. (Castro, 2015)

- Escalabilidad: la capacidad de adaptación y respuesta de un sistema con respecto al rendimiento del mismo a medida que aumentan de forma significativa el número de usuarios del mismo. Aunque parezca un concepto claro, la escalabilidad de un sistema es un aspecto complejo e importante del diseño. La escalabilidad está íntimamente ligada al diseño del sistema. Influye en el rendimiento de forma significativa. Si una aplicación está bien diseñada, la escalabilidad no constituye un problema. (Juntadeandalucia, 2020)

- Firewall: Un firewall o cortafuegos es un dispositivo de hardware o un software que nos permite gestionar y filtrar la totalidad de tráfico entrante y saliente que hay entre 2 redes u ordenadores de una misma red.
  Si el tráfico entrante o saliente cumple con una serie de Reglas que nosotros podemos especificar, entonces el tráfico podrá acceder o salir de nuestra red

u ordenador sin restricción alguna. En caso de no cumplir las reglas el tráfico entrante o saliente será bloqueado. (Carles, 2013)

- Gestión de riesgo: Debe poder cuantificar el riesgo y predecir su impacto en el proyecto. En consecuencia, el resultado es un riesgo aceptable o inaceptable.
  El primer paso en la Gestión de riesgo es el análisis de riesgo que tiene como propósito determinar los componentes de un sistema que requieren protección, sus vulnerabilidades que los debilitan y las amenazas que lo ponen en peligro, con el fin de valorar su grado de riesgo. (Erb, 2020)

- JSON (JavaScript Object Notation): es un formato ligero de intercambio de datos, que resulta sencillo de leer y escribir para los programadores y simple de interpretar y generar para las máquinas. (Barrera, 2020)

  JSON es un formato de texto completamente independiente de lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores, entre ellos:

  C
  C++
  C#
  Java
  JavaScript
  Perl
  Python
  Entre otros

- Kali Linux: Es una distribución de Linux avanzada para pruebas de penetración y auditorías de seguridad. Es una reconstrucción de BackTrack Linux, se adhiere al estándar del entorno Debían y contiene más de 300 herramientas de pruebas de penetración.
  Kali Linux trae preinstalados más de 600 programas incluyendo Nmap (un escáner de puertos), Wireshark (un sniffer), John the Ripper (un crackeador de passwords) y la suite Aircrack-ng (software para pruebas de seguridad en redes inalámbricas). Kali puede ser usado desde un Live CD, live-usb y también puede ser instalada como sistema operativo principal. (Wiki, 2020)

- Malware: Malware es un software que realiza tareas maliciosas en un dispositivo o red, como corromper datos o tomar el control de un sistema.
  Malware o “software malicioso” es un término amplio que describe cualquier programa o código malicioso que es dañino para los sistemas.
  El malware hostil, intrusivo e intencionadamente desagradable intenta invadir, dañar o deshabilitar ordenadores, sistemas informáticos, redes, tabletas y dispositivos móviles, a menudo asumiendo el control parcial de las operaciones de un dispositivo. Al igual que la gripe, interfiere en el funcionamiento normal. (Malwarebytes, 2020)

- NoSQL: Están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala. (Amazon, 2020)

- Tabla: Las tablas son objetos de base de datos que contienen todos sus datos. En las tablas, los datos se organizan con arreglo a un formato de filas y columnas, similar al de una hoja de cálculo. Cada fila representa un registro único y cada columna un campo dentro del registro. Por ejemplo, en una tabla que contiene los datos de los empleados de una compañía puede haber una fila para cada empleado y distintas columnas en las que figuran detalles de los mismos, como el número de empleado, el nombre, la dirección, el puesto que ocupa y su número de teléfono particular. (Microsoft, 2019)

- Redis: Redis es un almacén de estructura de datos en memoria de código abierto (licencia BSD), que se utiliza como agente de base de datos, caché y mensaje. Admite estructuras de datos como cadenas, hashes, listas, conjuntos, conjuntos ordenados con consultas de rango, mapas de bits, hiper blogs, índices geoespaciales con consultas de radio y flujos. Redis tiene replicación incorporada, secuencias de comandos Lua, desalojo de LRU, transacciones y diferentes niveles de persistencia en el disco, y proporciona alta disponibilidad a través de Redis Sentinel y particionamiento automático con Redis Cluster. (Redis, 2020)

- Wireshark: La herramienta intercepta el tráfico y lo convierte en un formato legible para las personas. Esto hace que sea más fácil identificar qué tráfico está cruzando la red, con qué frecuencia y la latencia que hay entre ciertos saltos. Si bien Wireshark admite más de 2.000 protocolos de red, muchos de

ellos inusuales o antiguos, los profesionales encuentran una gran utilidad en el análisis de identidades IP. La mayoría de los paquetes son TCP, UPD e ICMP. (Cso, 2018)

5.2 Marco teórico

En el presente capítulo se expondrán las definiciones y temas referidos a los motores de bases de datos creados y los datos, partiendo que se deben definir ciertas características tales como la concurrencia, seguridad y distribución de los datos. Es sobre una de estas áreas presentadas que se desarrollara el proyecto.

Dato

Una definición tomada del sitio “concepto.de” presenta a los datos en la informática como “Los datos son, así, la información (valores o referentes) que recibe el computador a través de distintos medios, y que es manipulada mediante el procesamiento de los algoritmos de programación. Su contenido puede ser prácticamente cualquiera: estadísticas, números, descriptores, que por separado no tienen relevancia para los usuarios del sistema, pero que en conjunto pueden ser interpretados para obtener una información completa y específica.” (Raffino, 2020)

Tomado por el diccionario de Cambridge “información, especialmente hechos o números, recopilados para ser examinados y considerados y utilizados para ayudar a la toma de decisiones, o información en forma electrónica que puede ser almacenada y utilizada por una computadora” (Dictionary, 2020)

Como se puede apreciar la definición de Cambridge concuerdan que un dato debe ser manejado por una computadora lo cual puede ser referido por archivos de información o bien por los DBMS creados tanto en SQL como NoSQL, por esta razón la definición de Cambridge será la que se utilizará puesto que habla directamente sobre cómo se va a tratar a esos datos.

DBMS

Un DataBase Management System (DBMS) o sistema gestor de bases de datos está definido como:

“Es un paquete de software diseñado para definir, manipular, recuperar y administrar datos en una base de datos. Un DBMS generalmente manipula los datos

en sí, el formato de datos, los nombres de campo, la estructura de registros y la estructura de archivos. También define reglas para validar y manipular estos datos." (Techopedia, 2019)

"Es un software de sistema para crear y administrar bases de datos. Un DBMS permite a los usuarios finales crear, leer, actualizar y eliminar datos en una base de datos. El tipo más frecuente de plataforma de gestión de datos, el DBMS, esencialmente sirve como una interfaz entre bases de datos y usuarios finales o programas de aplicación, asegurando que los datos estén organizados de manera consistente y sean fácilmente accesibles." (Rouse, 2020)

Las principales funciones de un DBMS según Margaret Rouse en su artículo "Database Management System" son los siguientes:

- Extracción de datos e independencia.
- Seguridad de datos.
- Un mecanismo de bloqueo para acceso concurrente.
- Un controlador eficiente para equilibrar las necesidades de múltiples aplicaciones que utilizan los mismos datos.
- La capacidad de recuperarse rápidamente de fallas y errores, incluyendo la manera en que el sistema retoma sus actividades, brinda sus respectivos servicios después de apagarse y la capacidad de recuperación.
- Robustas capacidades de integridad de datos.
- Registro y auditoría de actividad.
- Acceso simple usando una API estándar.
- Procedimientos de administración uniformes de datos.

A continuación, se explicarán los componentes de un DBMS

Componentes de un DBMS:

Los componentes de un DBMS listados por el sitio "studytonight" son: (Studytonight, 2020)

- Hardware:

  Cuando decimos Hardware, nos referimos a computadora, discos duros, canales de E / S para datos y cualquier otro componente físico involucrado antes de que cualquier información se almacene con éxito en la memoria.

Cuando ejecutamos Oracle o MySQL en nuestra computadora personal, el disco duro de nuestra computadora, nuestro teclado con el que escribimos todos los comandos, la RAM de nuestra computadora y la ROM se vuelven parte del hardware DBMS.

- Software:

  Este es el componente principal, ya que este es el programa que controla todo. El software DBMS es más como un contenedor alrededor de la base de datos física, que nos proporciona una interfaz fácil de usar para almacenar, acceder y actualizar datos.

  El software DBMS es capaz de comprender el lenguaje de acceso a la base de datos e interpretarlo en comandos reales de la base de datos para ejecutarlos en la base de datos.

- Datos:

  Los datos son ese recurso, para el cual se diseñó DBMS. El motivo detrás de la creación de DBMS fue almacenar y utilizar datos.

  En una base de datos típica, los datos guardados por el usuario están presentes y los metadatos se almacenan.

  Los metadatos son datos sobre los datos. Esta es información almacenada por el DBMS para comprender mejor los datos almacenados en él.

  Por ejemplo: cuando almaceno mi nombre en una base de datos, el DBMS se almacenará cuando el nombre se haya almacenado en la base de datos, cuál es el tamaño del nombre, si se almacena como datos relacionados con otros datos, o es independiente, toda esta información es metadata.

- Procedimientos:

  Los procedimientos se refieren a instrucciones generales para usar un sistema de gestión de bases de datos. Esto incluye procedimientos para configurar e instalar un DBMS, para iniciar y cerrar sesión en el software DBMS, para administrar bases de datos, para realizar copias de seguridad, generar informes, etc.

- Idioma de acceso a la base de datos:

  Database Access Language es un lenguaje simple diseñado para escribir comandos para acceder, insertar, actualizar y eliminar datos almacenados en cualquier base de datos.

  Un usuario puede escribir comandos en el lenguaje de acceso a la base de datos y enviarlo al DBMS para su ejecución, que luego el DBMS traduce y ejecuta.

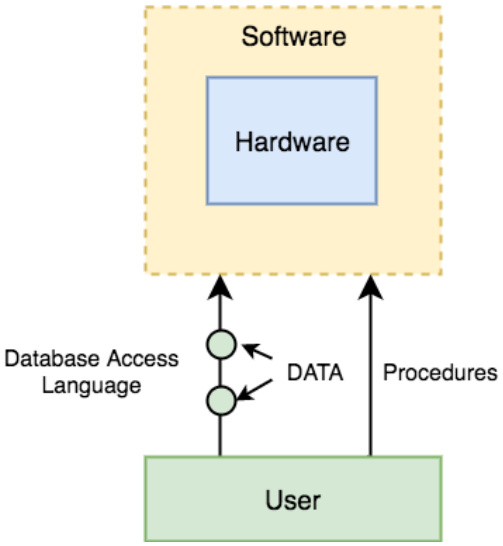
  El usuario puede crear nuevas bases de datos, tablas, insertar datos, buscar datos almacenados, actualizar datos y eliminar los datos utilizando el idioma de acceso.

- Motor de base de datos.

  El servicio principal para almacenar, procesar y asegurar datos, proporciona acceso controlado y procesamiento de transacciones rápido para satisfacer los requisitos de las aplicaciones de consumo de datos más exigentes. A menudo se usa para crear bases de datos relacionales para procesamiento de transacciones en línea o datos de procesamiento analítico en línea.

De igual manera el sitio muestra la siguiente figura la cual brinda una idea general del esquema de una base de datos

Ilustración 1 - Esquema de un sistema de base de datos



(Studytonight, 2020)

Como se puede observar en la ilustración 1 la manera adecuada de interactuar con los DBMS es a través de una aplicación para el usuario final, la misma que realiza una consulta, el cual en el DBMS se procesa y posteriormente se accede a los datos con la consulta procesada y finalmente se obtiene la información o se realiza cualquiera de los procesos que dicha consulta anteriormente solicitó.

DBMS SQL

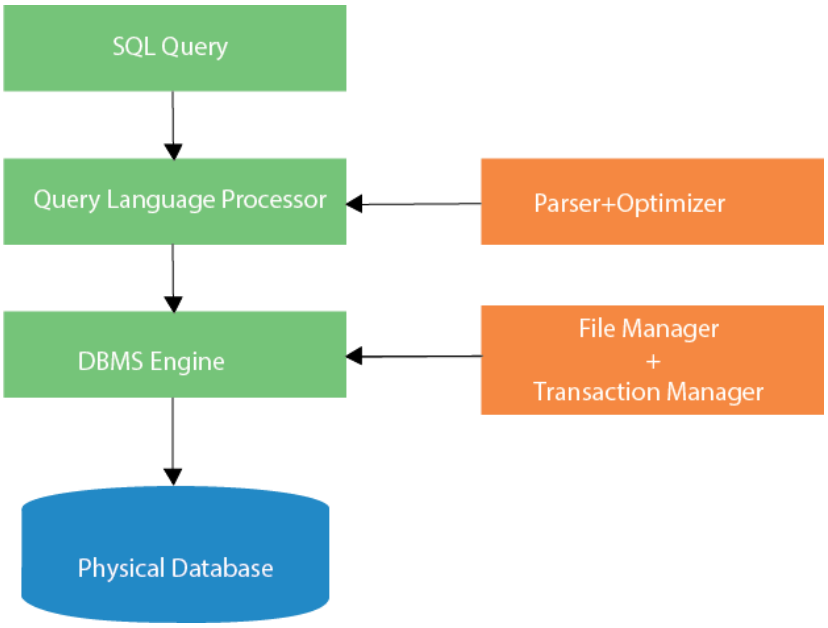
Los DBMS SQL por lo general son considerados RDBMS o Sistema de gestión de bases de datos relacional (SGBDR) puesto que se define como “SQL significa lenguaje de consulta estructurado. Se utiliza para almacenar y gestionar datos en el sistema de gestión de bases de datos relacionales (RDMS). Es un lenguaje estándar para el sistema de base de datos relacional. Permite al usuario crear, leer, actualizar y eliminar bases de datos y tablas relacionales.” (Java T Point, 2011)

El sitio “SQLcourse” señala que una DBMS SQL es “SQL se utiliza para comunicarse con una base de datos. Según ANSI (American National Standards Institute), es el lenguaje estándar para los sistemas de gestión de bases de datos relacionales. Las sentencias SQL se utilizan para realizar tareas como actualizar datos en una base de datos o recuperar datos de una base de datos.” (SQLcourse, 2019)

“SQL se puede utilizar para insertar, buscar, actualizar y eliminar registros de la base de datos. SQL puede realizar muchas otras operaciones, incluida la optimización y el mantenimiento de bases de datos. Las bases de datos relacionales como MySQL Database, Oracle, Ms SQL server, Sybase, etc. utilizan SQL.” (Guru99, 2020)

La idea del funcionamiento de estos DBMS se representa en la siguiente figura:

Ilustración 2 – Interacción SQL en bases de datos



(Java T Point, 2011)

Como se observa en la ilustración 2 el proceso correcto de interacción con una base de datos es:

1. Cuando se ejecuta un comando SQL o query para cualquier RDBMS, el sistema determina la mejor manera de llevar a cabo la solicitud y el motor SQL determina cómo interpretar la tarea.
2. En el proceso, se incluyen varios componentes. Estos componentes pueden ser motor de optimización, motor de consultas, despachador de consultas, clásico, etc.
3. Todas las consultas que no son SQL son manejadas por el motor de consultas clásico, pero el motor de consultas SQL no maneja archivos lógicos.

DBMS NoSQL

Un DBMS NoSQL es definido por Amazon como “Están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala.” (Amazon, 2020)

Porque se crearon los DBMS NoSQL por MongoDB “NoSQL abarca una amplia variedad de tecnologías de bases de datos diferentes que se desarrollaron en respuesta a las demandas presentadas en la creación de aplicaciones modernas:

- Los desarrolladores están trabajando con aplicaciones que crean volúmenes masivos de nuevos tipos de datos que cambian rápidamente: datos estructurados, semi estructurados, no estructurados y polimórficos.

- Atrás quedó el ciclo de desarrollo de la cascada de doce a dieciocho meses. Ahora los equipos pequeños trabajan en sprints ágiles, iterando rápidamente y presionando código cada semana o dos, algunas incluso varias veces al día. Las aplicaciones que alguna vez sirvieron a un público finito ahora se entregan como servicios que deben estar siempre activos, accesibles desde muchos dispositivos diferentes y escalados globalmente a millones de usuarios.” (MongoDB, 2020)

En la página web oficial de MongoDB se expone que existen 4 tipos de bases de datos que son:

- Las bases de datos de documentos emparejan cada clave con una estructura de datos compleja conocida como documento. Los documentos pueden contener pares diferentes de clave-valor, o pares de claves-matriz, o incluso documentos anidados.

- Los almacenes de gráficos se utilizan para almacenar información sobre redes de datos, como las conexiones sociales. Las tiendas de gráficos incluyen Neo4J y Giraph.

- Los almacenes de valores clave son las bases de datos NoSQL más simples. Cada elemento individual en la base de datos se almacena como un nombre de atributo (o 'clave'), junto con su valor. Ejemplos de tiendas de valores clave son Riak y Berkeley DB. Algunos almacenes de valores clave, como Redis,

permiten que cada valor tenga un tipo, como 'entero', que agrega funcionalidad.

- Los almacenes de columnas anchas como Cassandra y HBase están optimizados para consultas sobre grandes conjuntos de datos y almacenan columnas de datos juntos, en lugar de filas
  (MongoDB, 2020)

MongoDB

El sitio “Guru99” explica detalladamente lo que es el gestor de base de datos y hace hincapié en que es” MongoDB es una base de datos NoSQL orientada a documentos utilizada para el almacenamiento de datos de alto volumen. En lugar de usar tablas y filas como en las bases de datos relacionales tradicionales, MongoDB hace uso de colecciones y documentos. Los documentos consisten en pares clave-valor que son la unidad básica de datos en MongoDB. Las colecciones contienen conjuntos de documentos y funciones que son equivalentes a las tablas de bases de datos relacionales. MongoDB es una base de datos que salió a la luz a mediados de la década de 2000.” (Guru99, 2020)

Las características que más resaltan en MongoDB son:

1. Cada base de datos contiene colecciones que a su vez contienen documentos. Cada documento puede ser diferente con un número variable de campos. El tamaño y el contenido de cada documento pueden ser diferentes entre sí.
2. La estructura del documento está más en línea con la forma en que los desarrolladores construyen sus clases y objetos en sus respectivos lenguajes de programación. Los desarrolladores a menudo dirán que sus clases no son filas y columnas, sino que tienen una estructura clara con pares clave-valor.
3. Las filas (o documentos como se llama en MongoDB) no necesitan tener un esquema definido de antemano. En cambio, los campos se pueden crear sobre la marcha.
4. El modelo de datos disponible en MongoDB le permite representar relaciones jerárquicas, almacenar matrices y otras estructuras más complejas con mayor facilidad.

Apache Cassandra

La página “intellipaat” define a Apache Cassandra como “es un sistema de base de datos distribuido de código abierto extremadamente potente que funciona muy bien para manejar grandes volúmenes de registros distribuidos en múltiples servidores básicos. Se puede escalar fácilmente para satisfacer un aumento repentino de la demanda mediante la implementación de clústeres Cassandra de múltiples nodos y cumplir con los requisitos de alta disponibilidad, sin un solo punto de falla. Es una de las bases de datos NoSQL más eficientes disponibles en la actualidad. DataStax ofrece una distribución empaquetada gratuita de Apache Cassandra. Esto también incluye otras herramientas como Windows Installer, DevCenter y la documentación profesional de DataStax.” (Intellipaat, 2016)

El sitio oficial de “Apache Cassandra” reafirma que su motor de base de datos es el mejor hoy día “La base de datos Apache Cassandra es la elección correcta cuando necesita escalabilidad y alta disponibilidad sin comprometer el rendimiento. La escalabilidad lineal y la probada tolerancia a fallas en hardware básico o infraestructura de nube lo convierten en la plataforma perfecta para datos de misión crítica. El soporte de Cassandra para replicar en múltiples centros de datos es el mejor en su clase, brindando una latencia más baja para sus usuarios y la tranquilidad de saber que puede sobrevivir a las interrupciones regionales.” (Cassandra, 2016)

las características que más resaltan en Apache Cassandra son:

1. Tolerante A Fallos: Los datos se replican automáticamente en múltiples nodos para tolerancia a fallas. Se admite la replicación en múltiples centros de datos. Los nodos fallidos se pueden reemplazar sin tiempo de inactividad.
2. Durable: Cassandra es adecuada para aplicaciones que no pueden permitirse perder datos, incluso cuando un centro de datos completo deja de funcionar
3. Elástico: El rendimiento de lectura y escritura aumenta linealmente a medida que se agregan nuevas máquinas, sin tiempo de inactividad o interrupción de las aplicaciones.
4. Eficiente: Cassandra constantemente supera a las populares alternativas NoSQL en puntos de referencia y aplicaciones reales, principalmente debido a elecciones arquitectónicas fundamentales.

Redis

La página oficial de “Redis” define a su motor de bases de datos como “Redis es un almacén de estructura de datos en memoria de código abierto (licencia BSD), que se utiliza como agente de base de datos, caché y mensaje. Admite estructuras de datos como cadenas, hashes, listas, conjuntos, conjuntos ordenados con consultas de rango, mapas de bits, hiper blogs, índices geoespaciales con consultas de radio y flujos. Redis tiene replicación incorporada, secuencias de comandos Lua, desalojo de LRU, transacciones y diferentes niveles de persistencia en el disco, y proporciona alta disponibilidad a través de Redis Sentinel y particionamiento automático con Redis Cluster. ” (Redis, 2020)

Amazon proporciona un servicio que afirma que el uso adecuado para este gestor de bases de datos es en el área de Android ya que” Gracias a su velocidad y facilidad de uso, Redis es una opción popular para aplicaciones web, móviles, de juegos, de tecnología publicitaria y de IoT que requieren el mejor desempeño de su clase. AWS proporciona compatibilidad con Redis mediante un servicio de base de datos totalmente gestionado y optimizado llamado Amazon ElastiCache para Redis, y además permite a los clientes ejecutar Redis en AWS EC2 administrado por ellos mismos.” (Amazon, 2020)

las características que más resaltan en Redis son:

1. Desempeño increíblemente rápido: Todos los datos de Redis se encuentran en la memoria principal del servidor, a diferencia de la mayoría de sistemas de administración de bases de datos, que almacenan los datos en el disco o en SSD. Al eliminar la necesidad de acceder a discos, las bases de datos en memoria como Redis evitan los retrasos y pueden acceder a los datos con algoritmos más sencillos que utilizan menos instrucciones de la CPU. Las operaciones típicas tardan menos de un milisegundo en ejecutarse.
2. Estructuras de datos en memoria: Redis permite a los usuarios almacenar claves que se corresponden con diversos tipos de datos. El tipo de datos fundamental es una cadena, que puede componerse de texto o datos binarios y tener un tamaño de hasta 512 MB. Redis también admite listas de cadenas en el orden en el que se han agregado, conjuntos de cadenas sin ordenar, conjuntos clasificados ordenados por puntuación, hashes que almacenan una lista de campos y valores, e HyperLogLogs que cuentan los elementos únicos de un conjunto de datos. Con Redis, se puede almacenar en la memoria prácticamente cualquier tipo de datos.

3. Compatibilidad con su lenguaje de programación favorito: Los desarrolladores de Redis tienen a su disposición más de cien clientes de código abierto. Entre los lenguajes admitidos se encuentran Java, Python, PHP, C, C++, C#, JavaScript, Node.js, Ruby, R, Go y muchos otros.

Vulnerabilidad

Según el libro “Testing Guide Foreword” una vulnerabilidad es “una vulnerabilidad es una debilidad que puede ser explotada por un ataque cibernético para obtener acceso no autorizado o realizar acciones no autorizadas en un sistema informático. Las vulnerabilidades pueden permitir a los atacantes ejecutar código, acceder a la memoria de un sistema, instalar malware y robar, destruir o modificar datos confidenciales.”

Existen muchas definiciones de vulnerabilidad, estas son las más relevantes según el sitio “UpGuard”.

- Instituto Nacional de Estándares y Tecnología (NIST): Debilidad en un sistema de información, procedimientos de seguridad del sistema, controles internos o implementación que podría ser explotada o activada por una fuente de amenaza.

- ISO 27005: Una debilidad de un activo o grupo de activos que pueden ser explotados por una o más amenazas cibernéticas donde un activo es cualquier cosa que tenga valor para la organización, sus operaciones comerciales y su continuidad, incluidos los recursos de información que respaldan la misión de la organización.

- IETF RFC 4949: una falla o debilidad en el diseño, implementación u operación y administración de un sistema que podría explotarse para violar la política de seguridad del sistema.
  (Tunggal, 2020)

Ciber amenaza

Según el libro “Testing Guide Foreword” una ciber amenaza es “Una amenaza es cualquier cosa (un atacante externo malintencionado, un usuario interno, una inestabilidad del sistema, etc.) que puede dañar los activos propiedad de una

aplicación (recursos de valor, como los datos en una base de datos o en el sistema de archivos) mediante la explotación de una vulnerabilidad."

Según el sitio "UpGuard" las amenazas cibernéticas comunes incluyen:

- Malware: Malware es un software que realiza tareas maliciosas en un dispositivo o red, como corromper datos o tomar el control de un sistema.

- Spyware: El spyware es una forma de malware que se esconde en un dispositivo que proporciona información en tiempo real para compartir con su host, lo que les permite robar datos como datos bancarios y contraseñas.

- Ataques de phishing: el phishing es cuando un cibercriminal intenta atraer a las personas para que proporcionen datos confidenciales, como información de identificación personal (PII), datos bancarios y de tarjetas de crédito y contraseñas.

- Ataques distribuidos de denegación de servicio (DDoS): los ataques distribuidos de denegación de servicio tienen como objetivo interrumpir una red informática inundando la red con solicitudes superfluas para sobrecargar el sistema y evitar que se cumplan las solicitudes legítimas.

- Ransomware: el ransomware es un tipo de malware que niega el acceso a un sistema informático o datos hasta que se paga un rescate.

- Exploits de día cero: un exploit de día cero es una falla en el software, hardware o firmware que la parte o las partes responsables de reparar la falla desconocen.

- Amenazas persistentes avanzadas: una amenaza persistente avanzada es cuando un usuario no autorizado obtiene acceso a un sistema o red y permanece allí sin ser detectado durante un período prolongado de tiempo.

- Troyanos: un troyano crea una puerta trasera en su sistema, lo que permite al atacante obtener el control de su computadora o acceder a información confidencial.

- Ataques de limpiador: un ataque de limpiador es una forma de malware cuya intención es borrar el disco duro de la computadora que infecta.

- Robo de dinero: los ataques cibernéticos pueden obtener acceso a números de tarjetas de crédito o cuentas bancarias para robar dinero.

- Manipulación de datos: la manipulación de datos es una forma de ciberataque que no roba datos, sino que tiene como objetivo cambiar los datos para dificultar el funcionamiento de una organización.

- Destrucción de datos: la destrucción de datos es cuando un atacante cibernético intenta eliminar datos.

- Descargas drive-by: Un ataque de descarga drive-by es una descarga que ocurre sin el conocimiento de una persona que a menudo instala un virus informático, spyware o malware.

- Malvertising: Malvertising es el uso de publicidad en línea para difundir malware.

- Software malicioso: el software malicioso es un malware disfrazado de software real.

- Software sin parchear: el software sin parchear es un software que tiene una debilidad de seguridad conocida que se ha corregido en una versión posterior pero que aún no se ha actualizado.
  (Tunggal, 2020)

Common Vulnerabilities and Exposures

El sitio “Improving Security Together” sintetiza que el CVSS” es un marco abierto para comunicar las características y la gravedad de las vulnerabilidades de software. CVSS consta de tres grupos métricos: base, temporal y ambiental. El grupo Base representa las cualidades intrínsecas de una vulnerabilidad que son constantes a lo largo del tiempo y en los entornos de usuario, el grupo Temporal refleja las características de una vulnerabilidad que cambian con el tiempo y el grupo Ambiental representa las características de una vulnerabilidad que son exclusivas de un usuario medio ambiente. Las métricas base producen una puntuación que va de 0 a 10, que luego se puede modificar al calificar las métricas temporales y ambientales. Una puntuación CVSS también se representa como una cadena de

vectores, una representación textual comprimida de los valores utilizados para derivar la puntuación.” (First, 2020)

CVSS se compone del grupo de métrica Base:

- El grupo de métricas Base representa las características intrínsecas de una vulnerabilidad que son constantes a lo largo del tiempo y en los entornos de los usuarios. Se compone de dos conjuntos de métricas: las métricas de explotación y las métricas de impacto.
    - Las métricas de explotación reflejan la facilidad y los medios técnicos por los cuales se puede explotar la vulnerabilidad. Es decir, representan características de lo que es vulnerable, a lo que nos referimos formalmente como el componente vulnerable.
    - Las métricas de impacto reflejan la consecuencia directa de una explotación exitosa y representan la consecuencia de lo que sufre el impacto, al que nos referimos formalmente como el componente afectado.

Según análisis realizados y ofrecidos en la página del NIST y NVD que son organizaciones gubernamentales de tecnología y vulnerabilidades, de estados unidos se registraron vulnerabilidades en el gestor de bases de datos MongoDB que fueron calificados por medio del sistema de puntaje CVSS y se obtuvo como calificaciones 3.3, 6.4 y 7.5 (CVE-2020-11499, CVE-2020-7922 y CVE-2020-1929) en una escala de seguridad del 0 al 10 (donde cero no representa amenazas y 10 una amenaza crítica) donde los criterios de evaluación son:

1. Vector de ataque:
    a. Un vector de ataque es cualquier vía de acceso que un hacker puede usar para llevar a cabo un ciberataque, ya sea para acceder a una base de datos a través de una vulnerabilidad técnica o para obtener las credenciales de un usuario para posteriormente iniciar sesión en la red de alguna empresa, de esta manera se podrá robar o dañar los datos del sistema.
    b. Local (L): Una vulnerabilidad explotable solo con acceso local requiere al atacante tener ya sea acceso físico al sistema vulnerable o una cuenta local.

c. Red Adyacente (A): Una vulnerabilidad explotable con un acceso de red adyacente requiere al atacante tener acceso ya sea al dominio de difusión o colisión del software vulnerable.
d. Red (N): Una vulnerabilidad explotable con acceso a red significa que el software vulnerable está en la pila de red y el atacante no requiere acceso a la red local o acceso local.

2. Complejidad de ataque:
    a. Un ataque de complejidad algorítmica es un ataque de agotamiento de recursos que aprovecha el rendimiento que en el peor de los casos para un algoritmo de fondo resulta en el agotamiento de los recursos del servidor, esto surge porque la mayoría de los desarrolladores prueben sus algoritmos para el rendimiento promedio de casos aleatorios, comparando con los tipos de entradas que proporciona un usuario típico.
    b. Alta (H): La parte atacante debe tener privilegios elevados o burlar los sistemas adicionales en adición al sistema atacante.
    c. Media (M): La parte está limitada a un grupo de sistemas o usuarios con algún nivel de autorización, posiblemente no confiable.
    d. Baja (L): El producto afectado requiere típicamente acceso a un amplio rango de sistemas y usuarios, posiblemente anónimos o no confiables.

3. Privilegios requeridos:
    a. Un privilegio requerido señala el nivel que deberá tener un atacante antes de explotar con éxito la vulnerabilidad de algún sistema. El puntaje base es mayor si no se requieren privilegios.
    b. Ninguno (N): No se requiere autenticación para explotar la vulnerabilidad.
    c. Baja (L): La vulnerabilidad requiere al atacante registrar su ingreso en el sistema.
    d. Alta (H): Explotar la vulnerabilidad requiere la autenticación del atacante dos o más veces, aún si las mismas credenciales son usadas cada vez.

4. Interacción del usuario
    a. La interacción del usuario representa:
    b. Requerido (R): Una puntuación alta para cuando un usuario no requiere ninguna acción para que un ataque se ejecute con éxito.

c. Ninguna (N): Una puntuación más baja cuando se requiere cualquier tipo de interacción del usuario.

5. Alcance:
    a. El alcance se refiere al grupo de privilegios que se caracterizan por una autoridad informática al dar acceso a los recursos informáticos. Estos privilegios se designan según una técnica de aprobación e identificación.
    b. Sin cambios (U): El componente afectado y el componente vulnerable son iguales. Los recursos afectados están controlados por la misma autoridad.
    c. Cambiado (C): El componente afectado y el componente vulnerable son diferentes. La misma autoridad no controla los recursos afectados.

6. Confidencialidad:
    a. La confidencialidad es una métrica que limita el acceso a la información y revela información solo a usuarios autorizados. Además, evita la divulgación de información a usuarios no autorizados.
    b. Alto (H): Todos los recursos del componente afectado se revelan al atacante debido a la pérdida total de confidencialidad.
    c. Bajo (L): El atacante no puede controlar la información restringida que se obtiene. Es posible el acceso a algunos archivos del sistema, pero el atacante no tiene control sobre lo obtenido, o el alcance y la pérdida está limitada.
    d. Ninguna (N): Sin pérdida de confidencialidad.

7. Integridad:
    a. El impacto de integridad mide la verdadera naturaleza de la información y cuanto se puede confiar en ella. La explotación exitosa de la vulnerabilidad se mide a través del impacto en la integridad.
    b. Alto (A): Pérdida total de integridad o protección. El atacante puede alterar cualquier archivo.
    c. Bajo (L): El atacante puede modificar un archivo, pero el atacante no tiene control sobre lo modificable, o el alcance sobre lo afectado es limitado.
    d. Ninguno (N): Sin pérdida de integridad.

8. Disponibilidad: Se refiere a la cantidad de recursos de información accesible.

a. Alto (H): El atacante puede negar el acceso total a los recursos en el componente afectado existiendo que haya un reinicio total de estos recursos afectados conllevando a la pérdida total de disponibilidad.
b. Bajo (B): Existen interrupciones o desempeño reducido en la disponibilidad del recurso. Los recursos parciales o completos solo están disponibles durante un periodo determinado.
c. Ninguna (N): Sin pérdida de disponibilidad.

Metodología de pentesting

El sitio de blogs de aportes investigativos “Bing” presenta que una metodología de pentesting” El Ethical Hacking, es de las especializaciones de seguridad informática más apetecidas por las grandes empresas a nivel mundial, todos los días crece la necesidad de tener personas con los suficientes conocimientos en estas áreas, para contrarrestar los ataques de la creciente comunidad de hackers, la cual tiene mayor representación en Asia y el Medio Oriente, pero que está regada por todo el mundo. Es muy delgada la línea entre un hacker de sombrero blanco y un hacker de sombrero negro, a nivel de conocimientos ambos tienen la capacidad de reconocer vulnerabilidades y/o fallos en sistemas, para sacar provecho de la situación, el hacker ético tiene como misión explotar estas vulnerabilidades y reportar las mismas, el fin nunca es el sacar provecho económico de la situación, por lo contrario el objetivo es hacer recomendaciones y/o diseñar controles para la mejora del sistema.”(Blogger, 2013)

Las metodologías más usadas en pentesting son:

OWASP (Open Web Application Security Project): Con el objetivo de hallar vulnerabilidades se trabajará sobre un conjunto de normas ya establecidas que propone OWASP, organización que apoya proyectos de software por medio del apoyo de seguridad.

Las bases de datos NoSQL proporcionan restricciones de consistencia más flexibles que las bases de datos SQL tradicionales. Al requerir menos restricciones relacionales y verificaciones de consistencia, las bases de datos NoSQL a menudo ofrecen beneficios de rendimiento y escala. Sin embargo, estas bases de datos siguen siendo potencialmente vulnerables a los ataques de inyección, incluso si no están utilizando la sintaxis SQL tradicional. Debido a que estos ataques de inyección NoSQL pueden ejecutarse dentro de un lenguaje de procedimiento, en lugar de en

el lenguaje SQL declarativo, los posibles impactos son mayores que la inyección SQL tradicional, algunas de las características más representativas de OWASP son: (Blogger, 2013)

- Pruebas de firma digital de aplicaciones Web.
- Comprobaciones del sistema de autenticación.
- Pruebas de Cross Site Scripting.
- Inyección XML
- Inyección SOAP
- HTTP Smuggling
- SQL Injection
- LDAP Injection
- Polución de Parámetros

OSSTMM (Open-Source Security Testing Methodology Manual): Metodología que propone un proceso de evaluación de una serie de áreas que refleja de manera fiel los niveles de seguridad presentes en la infraestructura que va a ser auditada, a estos niveles de seguridad se le denominan comúnmente “Dimensiones de Seguridad” y normalmente consiste en analizar los siguientes factores:
(Blogger, 2013)

- Visibilidad
- Acceso
- Confianza
- Autenticación
- Confidencialidad
- Privacidad
- Autorización
- Integridad
- Seguridad
- Alarma

ISSAF (Information Systems Security Assessment Framework): Marco metodológico de trabajo desarrollado por la OISSG que permite clasificar la información de la evaluación de seguridad en diversos dominios usando diferentes criterios de prueba. Algunas de las características más representativas de ISSAF son:
(Blogger, 2013)

- Brinda medidas que permiten reflejar las condiciones de escenarios reales para las evaluaciones de seguridad.

- Esta metodología se encuentra principalmente enfocada en cubrir los procesos de seguridad y la evaluación de los mismos para así obtener un panorama completo de las vulnerabilidades existentes.

- Permite el desarrollo de matriz de riesgo para verificar la efectividad en la implementación de controles.

CEH (Certified Ethical Hacker): Metodología de pruebas de seguridad desarrollada por el International Council of Electronic Commerce Consultants (EC-Council) algunas de las fases enunciadas en esta metodología son:
(Blogger, 2013)

- Obtención de Información.
- Obtención de acceso.
- Enumeración.
- Escala de privilegios.
- Report

6. ESTADO DEL ARTE

Con fin de contextualizar el presente proyecto como estado de arte se indaga los trabajos y proyectos relacionados a pruebas de seguridad en bases de datos no relacionales.

6.1 Tomado de: “MongoDB NoSQL Injection Analysis and Detection”

18 de agosto de 2016

Como primera instancia se encuentra (Hou, Qian, Li and Shi, 2016) hacen una indagación profunda respecto a las medidas de seguridad de las bases de datos no relacionales, en general esta investigación va enfocada a la base de datos MongoDB con aspectos tanto ataque como de defensa a nivel de código, usando los entornos de JavaScript y PHP realizarán una prueba experimental en inyecciones NoSQL.

Esta demostración la harán como un ataque de inyección de JavaScript del lado del servidor contra un sistema de base de datos NoSQL revela los datos privados del cliente, en este orden de ideas proceden a hacer una declaración de consulta de NoSQL, dentro de este la salida actual debe estar con el número ISBN 0763754891

Ilustración 3 - Declaración de consulta NoSQL

```
SQL: "SELECT * FROM users WHERE (ISBN =
'" + isbn_number + "');"

MongoDB: db.collection.find( { ISBN:
isbn_number } )
```

- **Ataque**

Introdujeron dos diferentes formas de inyección:

1. La primera de ellas es directamente por cuadros de entrada, es decir, suponiendo que el sistema está escrito por PHP y JavaScript el sistema comienza a buscar documento por documento y la salida de la información del libro que coincide con `find(array('$where' => $jss))` como la declaración de condición es verdadero este generará automáticamente la información del libro, a partir de aquí un pirata informático podría inyectar una pieza de código malicioso para que todas las respuestas sean ciertas a la hora de buscar documento por documento

Ilustración 4 – Ataque por cuadros de entrada

```
$unsearch = $_POST['name'];
$jss = "function() {return this.ISBN == '$unsearch';}";
$cursor = $users->find(array('$where' => $jss));
```

2. La segunda de ellas es inyectar por URL, suponiendo que el sistema está escrito en PHP y posteriormente el usuario haga clic en algún botón de enviar o buscar, la consulta comienza. Cuando el ISBN es decir su número coincide, muestra la información del libro actual

Ilustración 5 – Inyección a través de la URL

```
$search = array (
            'ISBN' => $_GET['isbn']
        );
$cursor = $users->find($search);
```

Es posible usar esta estrategia ya que la ejecución de MongoDB también acepta las notaciones $.
Básicamente respecto a las inyecciones en consulta su idea principal es que haga declaraciones de condición, estas siempre serán verdaderas ya que los piratas informáticos buscar documento por documento su resultado muestre toda la información de la base de datos

- **Defensa:**

En cuanto a defensa discuten cómo evitar la inyección de cuadros de entrada en sitios web, proporcionando un enfoque de detección a nivel de amenazas, de esta manera la inyección no podrá causar una fuga de datos

Análisis de defensa de MongoDB NoSQL:

Para las defensas de MongoDB existen dos formas, validación de entrada y declaración parametrizada. Para el primero se debe limitar lo que un usuario desea ingresar, en cuanto a la consulta de numero de ISBM deberán ser todos los números y no uno por uno de esta manera cuando el desarrollador está construyendo el software puede agregar una pieza de código JavaScript para limitar los cuadros de entrada

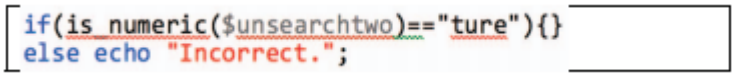
Ilustración 6 – Contramedida "Validación de entrada"

```
onkeypress="return
event.keyCode>=48&&event.keyCode<=57"
```

La segunda es declaración parametrizada se encarga en verificar y filtrar la variables, al momento de escribir una declaración de condición esta variable no es insertada directamente es decir la entrada del usuario no puede ser incrustada directamente en la declaración de condición, en cambio el

contenido de entrada del usuario deberá ser filtrado, con esta medida se podrá eliminar la mayoría de los ataques de inyección, un ejemplo sería un código que determine el carácter de una variable si esta contiene solo números o no, inmediatamente será comprobado por el número ISBN si el resultados de ésta es positivo pasará su valor, si no es rechazado, se representa de esta manera

Ilustración 7 – Contramedida "Declaración parametrizada"



Figuras tomadas de:
MongoDB NoSQL Injection Analysis and Detection
(Hou, Qian, Li and Shi, 2016)

6.2 Tomado de: "SECURITY ANALYSIS OF UNSTRUCTURED DATA IN NOSQL MONGODB DATABASE"

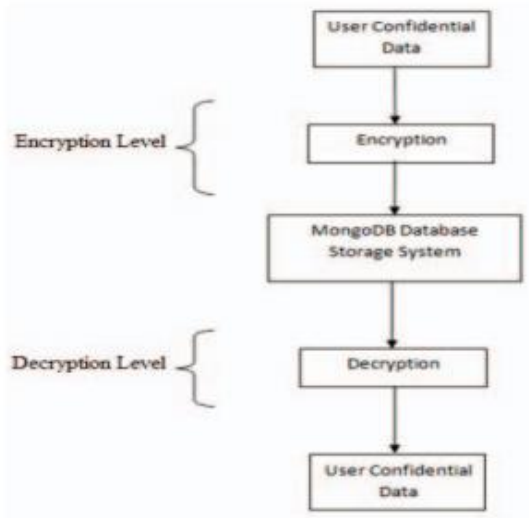
08 de febrero de 2018

Como segunda instancia se encuentra el artículo (KUMAR and GARG, 2017) en todo su trabajo usan el atributo de calidad de confidencialidad como base para el desarrollo de su proyecto, usando técnicas criptográficas simétricas desean gestionar el almacenamiento de los datos de un restaurante para bases de datos no relacionales como MongoDB, para apoyarse en su investigación usan tres técnicas criptográficas como AES, DES y blowfish que es un generador aleatorio de claves de cifrado, de esta manera vulneraron la seguridad de la base de datos de MongoDB recuperando sus datos.

Cómo desean analizar datos no estructurados para analizar en tiempo real la complejidad de usar diferentes técnicas criptográficas con MongoDB para diferentes tamaños de datos del restaurante y estos tienen la forma del formato JSON.

La estructura de cifrado que usaron para los datos en MongoDB sigue una serie de pasos donde primero los datos que confidenciales están encriptados antes de estar almacenados en la base de datos y después son descifrados accediendo a la base de datos, en general su estructura fue la siguiente:

Ilustración 8 – Estructura de cifrado



NoSQL MongoDB tiene muchas vulnerabilidades de seguridad, pero a pesar de esto obtuvieron que AES es mucho más seguro que otros porque la clave de longitud es de 128 bits según la sustitución y permutación de red, pero esto no quiere decir que no sea posible el ataque en Blowfish y DES. size representa el tiempo empleado en el proceso de encriptado para diferentes conjuntos de tamaño de datos, pero este tiempo de ejecución de cifrado es tomado por el promedio de cada una de las tres técnicas de cifrado al ser ejecutadas 10 veces en el código de recuperación de datos en la base de datos de MongoDB, El resultado fue el siguiente:

Size KB	Data Set	AES (128)		DES (64)		Blowfish (64)	
		Encryption (ms)	Decryption (ms)	Encryption (ms)	Decryption (ms)	Encryption (ms)	Decryption (ms)
365	SetA	139	97	153	119	43	31
866	SetB	212	153	292	204	69	46
1207	SetC	249	190	389	284	98	53
1582	SetD	298	246	449	303	108	73
1984	SetE	354	286	505	362	139	87

Tabla 1 - Análisis de datos técnica de cifrado

Figuras tomadas de:
SECURITY ANALYSIS OF UNSTRUCTURED DATA IN NOSQL MONGODB DATABASE
(KUMAR and GARG, 2017)

6.3 Tomado de: “Analysis Of NoSQL Database Vulnerabilities”

8 de mayo de 2018

El tercer artículo científico fue tomado de (GUPTA, SINGH and TOMAR, 2018) el cual discute sobre los modelos de datos NoSQL y describe sus numerosas características. Después de esta demostración, analiza las vulnerabilidades de seguridad, los diferentes mecanismos de ataque en las bases de datos NoSQL y las técnicas de mitigación.

Vulnerabilidades NoSQL y mecanismos de ataque:

Aunque NOSQL ofrece varias características, aún pueden existir vulnerabilidades que permiten operaciones arbitrarias en la base de datos.
Los atacantes pueden aprovechar las debilidades y pueden usarlas para explotar la base de datos y hacer que el sistema sea inseguro. Como las bases de datos almacenan información confidencial de las organizaciones, es importante hacerlas seguras.

Los principales mecanismos de los ataques de inyección NoSQL son:

- **Tautologías:** Una tautología se refiere a una expresión o una declaración condicional que siempre es verdadera. El objetivo de una inyección de tautología es apuntar a la parte condicional de una consulta para que la condición siempre sea verdadera en la evaluación. Debido a que el atacante puede ingresar al sistema y puede ejecutar acciones ilegales.

- **Consultas sindicales:** Esta es una técnica conocida en la que el ataque se realiza insertando una consulta de unión con algún contenido malicioso en un parámetro vulnerable. Esto lleva a una evaluación incorrecta de toda la declaración y cambia el conjunto de datos de resultados de la consulta original.

- **Consultas ilegales / lógicamente incorrectas:** En esta técnica, el atacante pasa algún parámetro no válido o consulta incorrecta a la base de datos y después de la evaluación, la base de datos devuelve un mensaje de error predeterminado. Los atacantes aprovechan esta vulnerabilidad e intentan obtener información sobre el back-end utilizando estas consultas lógicas.

- **Inyecciones de JavaScript:** La base de datos NoSQL presenta un tipo de nueva vulnerabilidad, la inyección de JavaScript. El uso de JavaScript puede proporcionar una superficie de ataque a los piratas informáticos, ya que

pueden realizar la inyección de código arbitrario de JavaScript para piratear el sistema y ejecutar la extracción o alteración ilegal de datos.

- **Inyección ciega NoSQL:** Aquí, el objetivo principal de un atacante es recopilar tanta información como sea posible sobre la base de datos y sus contenidos. En este ataque, los atacantes se centran principalmente en la respuesta del servidor para una condición verdadera y una condición falsa. Así, al hacer muchas preguntas verdaderas o falsas, los atacantes intentan extraer el contenido de la base de datos

Análisis y mitigación:

Mitigar el riesgo de seguridad es un gran problema en las bases de datos NoSQL. Hay varias formas a través de las cuales un atacante puede atacar el sistema. Para proteger el sistema de estos atacantes, ellos han propuesto numerosas técnicas. La técnica de mitigación la definieron en dos fases.

1. Desarrollo y pruebas: Para abordar completamente todo el problema, es necesario considerar todo el ciclo de vida de desarrollo de software. Para mitigar los riesgos de seguridad, también requiere centrarse en todos los aspectos mencionados a continuación:

    - **Conciencia:** La conciencia es la forma menos costosa de reducir el riesgo de seguridad. Se sugiere que todas las personas involucradas en el ciclo de vida del desarrollo deben tener una comprensión adecuada sobre las debilidades del sistema.

    - **Diseño:** Todos los aspectos de seguridad de una aplicación deben definirse en las primeras etapas. Esto asegurará una atención adecuada al trabajo incluso durante el ciclo de desarrollo

    - **Buena práctica de codificación:** La mayoría de los ataques se realizan debido a una mala desinfección. Un código debidamente validado puede reducir el riesgo de ataques. El uso de una sintaxis bien formada, un formato JSON fuerte, bibliotecas probadas, etc. minimiza el daño en el sistema.

    - **Aislamiento de privilegios:** Como las bases de datos NoSQL emplea autenticación y autorización y admiten el control de acceso basado en roles. Funcionan según el principio de privilegios mínimos. El aislamiento adecuado de privilegios reduce el riesgo de ataque en la base de datos.

- **Escaneo de seguridad:** Los desarrolladores deben ejecutar pruebas de seguridad dinámicas y estáticas con frecuencia. Esto ayudará a identificar las vulnerabilidades en el sistema de antemano. De esta manera, los errores pueden corregirse en el momento correcto

2. Monitoreo y detección de ataques: Incluso después de tener en cuenta todos los aspectos de seguridad anteriores, todavía existen vulnerabilidades en el sistema. Todos los días se introduce un nuevo vector de ataque, sobre el cual uno puede no saber en el momento del desarrollo. Por lo tanto, es necesario monitorear y defender el sistema en tiempo de ejecución. Los siguientes métodos pueden usarse para este propósito

    - **Cortafuegos de aplicaciones web:** Los firewalls se pueden usar para detectar ataques a nivel de red. Los WAF se utilizan para detectar transacciones HTTP maliciosas y flujos de datos HTTP. También se pueden agregar algunas reglas en WAF para detectar los ataques en el sistema de base de datos.

    - **Sistema de detección de intrusos:** IDS se puede utilizar para detectar comportamientos anormales en el sistema. Cada vez que hay un comportamiento inesperado, genera alertas e indica un ataque.

    - **Monitoreo de actividad de datos:** La herramienta de monitoreo de actividad se ha convertido en un requisito común para la protección de datos. Monitorean todas las actividades del sistema, crean un informe de auditoría, supervisan el acceso a la base de datos y generan alertas de seguridad. Por lo tanto, estas herramientas son útiles para detectar ataques en las bases de datos.

    - **Sistemas SIEM:** Los sistemas de información de seguridad y gestión de eventos (SIEM) ayudan a detectar ataques en la base de datos. Utilizan herramientas de inteligencia de amenazas para detectar la posibilidad de ataque en el sistema.

6.4 Tomado de: “NoSQL Injection Attack Detection in Web Applications Using RESTful Service”

06 marzo 2019

El cuarto artículo investigativo fue tomado de (EASSA, 2018) este documento presenta un servicio web RESTful independiente con un enfoque basado en capas

para detectar ataques de inyección NoSQL en aplicaciones web. El método propuesto se denomina DNIARS. DNIARS depende de la comparación de los patrones generados a partir de la estructura de instrucciones NoSQL en estado de código estático y estado dinámico. En consecuencia, el DNIARS puede responder a la aplicación web con la posibilidad de ataque de inyección NoSQL.

El DNIARS que los autores propusieron fue implementado en código simple PHP y donde pueda considerarse como un marco independiente que tiene la capacidad de responder a diferentes formatos como JSON, XML. Para evaluar su rendimiento, DNIARS lo probaron utilizando las herramientas de prueba más comunes para el servicio web RESTful.

Hay muchas bases de datos NoSQL que no son similares entre sí en cuanto a manipulación. Por el contrario, las bases de datos relacionales (como MS SQL Server, MySQL) usa SQL como lenguaje estándar para manipular bases de datos. Además, con múltiples idiomas en una aplicación web como (ASP, PHP), las instrucciones SQL se pueden insertar fácilmente dentro de ellas en un formato. Pero, por otro lado, las bases de datos NoSQL son diferentes en la forma de manipular las bases de datos.
Es aquí donde entra la investigación de este artículo científico ya que el gran desafío que se les presenta es que hay patrones que se pueden generar para la inyección NoSQL ataques dependiendo de la forma diferente de manipular cada tipo de bases de datos NoSQL. Es por ello que necesitan crear un modelo que pueda manejar todas estas formas de patrones al mismo tiempo.

Tabla 2 - Estado de ataque de inyección NoSQL

Estado	Estado Descripción
200	Ok - No se detectó ningún ataque de inyección NoSQL
400	Solicitud incorrecta: el ataque de inyección NoSQL ha sido detectado
404	no encontrado, tal vez significa el tipo de NoSQL base de datos no compatible o parámetros no válidos

Tabla obtenida de:
(EASSA, 2018)

Por lo general, hay tipos de ataques para cualquier sistema basado en red/internet. Estos ataques dependen principalmente de las características del entorno para las que está diseñada la aplicación informática (por ejemplo, aplicaciones web).

Existen muchas técnicas de aplicación web con ataques que se pueden utilizar de acuerdo con el tipo de cada Base de datos NoSQL que esté vinculada a la aplicación.

Con el modelo que los autores plantearon anteriormente discutirán con algunos ejemplos de ataque por inyección a las bases de datos NoSQL más comunes en la actualidad

Según el ranking de popularidad de las bases de datos más aceptados por las empresas se encuentra cuatro, estas son (MongoDB, Cassandra, Amazon y CouchDB) Los autores escogieron estas bases de datos NoSQL para confirmar que la probabilidad de ataque de inyección es diferente en técnica dependiendo del tipo de cada base de datos.

Ellos presentaron cuatro ejemplos de aplicaciones web que en sí son similares en lenguaje de función y programación, estos ejemplos se ejecutan después de enviar el inicio de sesión de usuario botón y la aplicación web verifica que ya han sido registrados en la base de datos NoSQL.

- Ejemplo 1: aplicación web conectada a MongoDB:

  ```
  $cont=$collection->findOne(array
  ('email'=>$email,'password'=> $pass));
  ```

  En el ejemplo 1, es posible inyectar un código en los datos enviados desde la página web de inicio de sesión. Esto podría, por ejemplo, verse así:

  ```
  http://127.0.0.1/phd/MongoDB/after_log.php?user=ahmed&pa
  ss[$ne]= 1&sbumit1=Submit
  ```

  Cuando "$ne" como se describe en la sección anterior, selecciona los documentos en los que el valor del campo no es igual a "1". Por lo tanto, el atacante tendrá la autoridad del usuario autorizado.

- Ejemplo 2: aplicación web conectada a Cassandra:

```
$users = $database->query('SELECT * FROM “reg_users” WHERE “username” = :usern' AND “password” = :passw', ['usern' => $user, 'passw' => $pass]);
```

En el ejemplo 2, la técnica de inyección es similar a lo que sucede en las bases de datos relacionales porque el lenguaje de consulta Cassandra CQL está cerca de SQL del lenguaje de consulta de base de datos relacional. Por lo tanto, el atacante puede introducir cualquier nombre de usuario y una contraseña de:

```
ali’; DROP COLUMNFAMILY ’users
```

El atacante puede eliminar completamente una tabla o COLUMNFAMILY incluyendo todos los datos dentro de ella y la consulta aparecerá de la siguiente manera:

```
('select * from reg_users where username = ali and password = ali'; drop columnfamily 'users', ['usern' =>, 'passw' => ali'; drop columnfamily 'users]).
```

- Ejemplo 3: aplicación web conectada a Amazon DynamoDB:

```
$response = $client->getItem(array (“TableName” => "usr_reg","Key" => array(“useremail” => array( Type:: STRING => $usrId),"password" => array( Type::STRING => $password))));
```

En el ejemplo 3, el atacante puede inyectar un código en los datos enviados desde la página web de inicio de sesión. Esto podría, por ejemplo, verse así: `http://127.0.0.1/phd/AmazonDynamoDB/after_log.php?user=ahmed&pass[$gt]= 1&sbumit1=Submit`

En Amazon DynamoDB, "$gt" es un operador utilizado para seleccionar los valores en los que el valor del documento o clave es mayor que el valor especificado. Por lo que el atacante puede acceder a la base de datos como un usuario autorizado.

- Ejemplo 4: aplicación web conectada a CouchDB:

```
sprintf('http://127.0.0.1:5984/test/?$reg_users['username']=%s&$reg_users['password']=%s', $user, $pass));
```

En el ejemplo 4 el atacante puede introducir cualquier nombre de usuario y una contraseña de

```
''or 1=1
```

Esto da como resultado la consulta URL de:

```
http://127.0.0.1/phd/CouchDB/after_log.php?user=test&pas
s=%27%27or+1%3D1&sbumit1=Submit
```

Por lo tanto, el atacante puede tener acceso a la base de datos como un usuario autorizado porque 1-1 siempre es true.

De acuerdo con los ejemplos anteriores, existe la posibilidad de atacar diferentes tipos de aplicaciones web que están conectadas a bases de datos NoSQL mediante la inyección de un código malicioso a través de diferentes técnicas.

7. METODOLOGIA

Para el desarrollo de este estudio se realizará una serie de fases para determinar el nivel de seguridad, de las bases de datos no relacionales, basándose en la identificación de vulnerabilidades.

7.1 Fase 1. Recolección de información

Inicialmente se hará la toma de datos o levantamiento de información realizando un estudio de las bases de datos no relacionales, cabe destacar que las bases de datos NoSQL son utilizadas en menor medida que las tradicionales bases de datos razón por la cual se deben presentar las características y funciones generales de las mismas.

Adicionalmente para facilitar el enfoque y comprensión del proyecto se emplearán sistemas gestores de bases de datos no relacionales específicos para este caso MongoDB, Redis y Apache Cassandra. Entre la información necesaria se encuentran también tipos de ataques o vulnerabilidades de seguridad que se pueden aplicar sobre estos sistemas además de las contramedidas impuestas por las compañías desarrolladoras e investigaciones realizadas sobre seguridad en bases de datos no relacionales. Una vez obtenidas y organizadas las vulnerabilidades se formulará un plan de pruebas que se aplicará sobre el programa.

Concretamente la recolección de información se enfoca en:

- Amenazas
- Vulnerabilidades
- Ataques
- Protección
- Estándares o normas
- Herramientas

7.2 Fase 2. Elaboración del plan de pruebas

El objetivo del plan de pruebas es organizar los ataques que se implementarán en el prototipo serán organizados por un formato que contendrá atributos de identificación, descripción, fecha, tipo de ataque ejecutado, resultado esperado, resultado obtenido y observaciones.

- Etapas
- Técnicas
- Herramientas
- Reporte

7.3 Fase 3. Implementación del prototipo y pruebas

Para la ejecución del plan de pruebas y para realizar los ataques será necesario implementar los prototipos cada uno con su respectivo gestor de bases de datos de tipo no relacional en una máquina virtual de manera que se encontrarán alojados en máquinas denominadas víctimas y la máquina atacante será otra máquina que contará con el sistema operativo de Kali.

7.4 Fase 4. Producción de informe

Finalmente se realizará el análisis de los resultados en un informe detallado con las pruebas que se hicieron junto a las fallas de seguridad encontradas para obtener una comparación entre los tres motores y se propondrán unas contramedidas con base en esos resultados y teniendo en cuenta estándares como lo son la ISO-27000 y la metodología de OWASP. Aunque existen fallas de hardware, software, desastre natural y errores humanos, el proyecto plantea trabajar únicamente con errores de software.

7.5 Herramientas

Es fundamental contar con el sistema operativo y el software del equipo y para precisar el funcionamiento de la base de datos no relacional hay que tener en cuenta su versión, con base a esto se utilizarán específicamente las siguientes herramientas:

7.5.1 Motor NoSQL

Mientras se diseña el prototipo, este deberá contar con una base de datos que almacene la información de los usuarios o información que se presente en la página. Adicionalmente cada prototipo contará con un gestor de bases de datos diferente.

7.5.2 Máquinas virtuales

Para realizar las pruebas se conectarán dos máquinas virtuales en donde una desempeña el papel de atacante y otra el de víctima. La metodología se desarrolla de esta manera para evitar realizar un ataque por medio de red que pueda registrarse como ilegal.

7.5.3 Kali (Herramientas de pentesting)

Se utilizará como sistema operativo la distribución de Linux llamada Kali debido a que está enfocada en utilizar herramientas que se utilizan para aplicar pruebas, auditorías y en seguridad informática.

7.5.4 Estándares y normas y buenas prácticas de seguridad

El proceso completo para el prototipo se desarrolla por medio de la metodología provista por OWASP se divide en fases de planear antes de iniciar, durante la definición y el diseño, durante el desarrollo, durante la implementación y mantenimiento.

Se utilizará la norma ISO-27000 para establecer estándares, normas de seguridad y buenas prácticas. Adicionalmente se consultará información que proveen las mismas compañías de los gestores utilizados en su respectiva documentación en sus páginas oficiales.

8. DESARROLLO DEL PROYECTO

8.1 Fase 1. Recolección de información.

Cumpliendo con la metodología del proyecto en su fase inicial habla sobre la recopilación de información de los principales ataques, vulnerabilidades y sus respectivas contramedidas, también se hizo el análisis de las más recientes vulnerabilidades de seguridad proporcionadas por la página del CVE para los gestores de bases de datos NoSQL MongoDB, Apache Cassandra y Redis.

8.1.1 Amenazas

Se hizo la recolección de las principales amenazas que han sido documentadas por parte de investigadores orientado al área de amenazas de seguridad en base de datos NoSQL.

Tabla 3 – Amenazas BD - NoSQL

Nombre	Descripción	Gestor	Fuente
Inyección	Ocurren cuando se envían datos que no son de confianza a un intérprete como parte de un comando o consulta.	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Owasp, 2017)
Tautologías	Una tautología es una expresión que siempre es cierta. Instintivamente, el objetivo de una inyección de tautología es apuntar a la parte de una consulta para que la condición siempre sea verdadera tras la evaluación	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Jonathan P, 2015)
Consultas ilegales o lógicamente incorrectas	El objetivo de este tipo de inyección es recopilar información sobre la base de datos. La causa aproximada de esta vulnerabilidad es que el DBMS proporciona demasiada información útil en el mensaje de error resultante. Esa información útil sería esquema de ataque a la base de datos y la versión de DBMS.	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Jonathan P, 2015)
Consultas de unión	El objetivo de este tipo de inyección es anexar una consulta adicional a la solicitud original para recuperar información de esa base de datos. Con el fin de utilizar eficazmente una consulta de unión, el atacante requiere	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Jonathan P, 2015)

	el conocimiento del esquema de base de datos. Si un atacante es capaz de determinar el tipo DBMS (tal vez a través de una consulta ilegal o lógicamente incorrecta), entonces el atacante podría aprovechar el conocimiento de las bases de datos.		
Consultas de Piggy-back	El objetivo de este tipo de inyección es insertar varias consultas en la base de datos. La consulta Piggy-back toma ventaja de DBMS que permiten que varias consultas que utilizan un símbolo especial (como un punto y coma) separen las consultas. Un atacante podría recuperar información adicional, agregar información a la base de datos, realizar un ataque de denegación de servicio, o incluso ejecutar procedimientos especiales almacenados en la DBMS.	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Jonathan P, 2015)
Autenticación rota	Las funciones relacionadas con la administración de sesiones a menudo se implementan sin protocolos establecidos de seguridad, lo que permite a los atacantes comprometer información para asumir las identidades de los usuarios.	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Owasp, 2017)
Exposición de datos sensibles	Puede ocurrir que no se protegen adecuadamente los datos confidenciales. Los atacantes pueden robar o modificar esos datos para cometer fraude de identidad. En este punto destaca la importancia de utilizar cifrados.	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Owasp, 2017)
Entidades externas XML (XXE)	Muchos procesadores XML antiguos o mal configurados evalúan las referencias de entidades externas dentro de los documentos XML. Las entidades externas se pueden utilizar para divulgar archivos internos mediante el controlador de URI de archivos, recursos compartidos de archivos internos, escaneo de puertos internos, ejecución remota de código y ataques de denegación de servicio.	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Owasp, 2017)
Control de acceso roto	Las restricciones sobre lo que pueden hacer los usuarios autenticados a menudo no se aplican correctamente.	MongoDB, Cassandra, Redis,	(Owasp, 2017)

	Los atacantes pueden explotar estas fallas para acceder a funciones y / o datos no autorizados, como acceder a las cuentas de otros usuarios, ver archivos confidenciales, modificar los datos de otros usuarios, cambiar los derechos de acceso, etc.	Riak, CouchDB, membase, neo4j.	
Mala configuración de seguridad	La configuración incorrecta de seguridad es el problema más común. Esto suele ser el resultado de configuraciones predeterminadas inseguras, configuraciones incompletas o ad hoc, almacenamiento en la nube abierta, encabezados HTTP mal configurados y mensajes de error detallados que contienen información confidencial. No solo todos los sistemas operativos, marcos, bibliotecas y aplicaciones deben configurarse de forma segura, sino que también deben actualizarse o actualizarse de manera oportuna.	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Owasp, 2017)
Scripting XSS entre sitios	Los defectos de XSS ocurren cuando una aplicación incluye datos que no son de confianza en una nueva página web sin la validación o el escape adecuados, o actualiza una página web existente con datos proporcionados por el usuario mediante una API de navegador que puede crear HTML o JavaScript. XSS permite a los atacantes ejecutar scripts en el navegador de la víctima que pueden secuestrar sesiones de usuarios, desfigurar sitios web o redirigir al usuario a sitios maliciosos.	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Owasp, 2017)
Deserialización insegura	Es un caso de ejecución remota de código. Incluso si las fallas de deserialización no dan como resultado la ejecución remota de código, se pueden usar para realizar ataques, incluidos ataques de reproducción, ataques de inyección y ataques de escalada de privilegios.	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Owasp, 2017)
Uso de componentes con vulnerabilidades conocidas	Los componentes, como bibliotecas, marcos y otros módulos de software, se ejecutan con los mismos privilegios que la aplicación. Si se explota un componente vulnerable, dicho ataque puede facilitar la pérdida de datos o la	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Owasp, 2017)

	toma de control del servidor. Las aplicaciones y API que utilizan componentes con vulnerabilidades conocidas pueden socavar las defensas de las aplicaciones y permitir varios ataques e impactos.		
Registro y monitoreo insuficientes	El registro y la supervisión insuficientes, junto con una integración faltante o ineficaz con la respuesta a incidentes, permiten a los atacantes atacar aún más los sistemas, mantener la persistencia, cambiar a más sistemas y manipular, extraer o destruir datos. La mayoría de los estudios de infracciones muestran que el tiempo para detectar una infracción es de más de 200 días, generalmente detectados por partes externas en lugar de procesos internos o monitoreo.	MongoDB, Cassandra, Redis, Riak, CouchDB, membase, neo4j.	(Owasp, 2017)

8.1.2 Vulnerabilidades de seguridad según el estándar CVE

CVE details es una lista de información registrada sobre vulnerabilidades de seguridad conocidas, en la que cada referencia tiene un número de identificación CVE-ID, descripción de la vulnerabilidad, que versiones del software están afectadas, posible solución al fallo (si existe) o como configurar para mitigar la vulnerabilidad de algún gestor de bases de datos, el proceso de investigación se hará con base a los informes de seguridad que hayan sido registrados recientemente para los gestores de bases de datos NoSQL MongoDB, Apache Cassandra y Redis.

- MongoDB

Tabla 4 – CVE Details MongoDB

Detalles de vulnerabilidad	Descripción	Puntaje CVSS y fecha de publicación	Fuente
CVE-2019-2390	Un usuario o programa sin privilegios en Microsoft Windows que puede crear archivos de configuración de OpenSSL en una ubicación fija puede hacer que los programas de utilidad enviados con versiones del servidor MongoDB inferiores a 4.0.11, 3.6.14 y 3.4.22 ejecuten código definido por el atacante como el usuario que ejecuta la utilidad.	Puntaje CVSS 6.8 Fecha de publicación: 2019-08-30	(CVE, D., *2390*)

CVE-2019-2389	El alcance incorrecto de las operaciones de eliminación en los scripts de inicio SysV empaquetados del servidor MongoDB permiten a los usuarios con acceso de escritura al archivo PID insertar PID arbitrarios para ser eliminados cuando el usuario root detiene el proceso MongoDB a través de SysV init. Este problema afecta a: versiones de MongoDB Inc. MongoDB Server v4.0 anteriores a la 4.0.11; versiones v3.6 anteriores a 3.6.14; Versiones v3.4 anteriores a 3.4.22.	Puntaje CVSS 3.3 Fecha de publicación: 2019-08-30	(CVE, D., *2389*)
CVE-2019-2386	Después de la eliminación del usuario en MongoDB Server, la invalidación incorrecta de las sesiones de autorización permite que la sesión de un usuario autenticado persista y se combine con nuevas cuentas, si esas cuentas reutilizan los nombres de las eliminadas. Este problema afecta a: versiones de MongoDB Inc. MongoDB Server v4.0 anteriores a la 4.0.9; versiones v3.6 anteriores a 3.6.13; Versiones v3.4 anteriores a 3.4.22.	Puntaje CVSS 6.0 Fecha de publicación: 2019-08-06	(CVE, D., *2386*)

- Apache Cassandra:

Tabla 5 – CVE Details Apache Cassandra

Detalles de vulnerabilidad	Descripción	Puntaje CVSS y fecha de publicación	Fuente
CVE-2018-8016	La configuración predeterminada en Apache Cassandra 3.8 a 3.11.1 vincula una interfaz JMX / RMI no autenticada a todas las interfaces de red, lo que permite a los atacantes remotos ejecutar código Java arbitrario a través de una solicitud RMI. Este problema es una regresión de CVE-2015-0225. La regresión se introdujo en https://issues.apache.org/jira/browse/CASSANDRA-12109. La corrección para la regresión se implementa en https://issues.apache.org/jira/browse/CASSANDRA-14173. Esta corrección está incluida en la versión 3.11.2 de Apache Cassandra.	Puntaje CVSS 7.5 Fecha de publicación: 2018-06-28	(CVE, D., *8016*)
CVE-2015-0225	La configuración predeterminada en Apache Cassandra 1.2.0 a 1.2.19, 2.0.0 a 2.0.13 y 2.1.0 a 2.1.3 vincula una interfaz JMX / RMI no autenticada a todas las interfaces de red, lo que permite a los atacantes remotos	Puntaje CVSS 7.5	(CVE, D., *0225*)

	ejecutar Java arbitrariamente. código a través de una solicitud RMI.	Fecha de publicación: 2015-04-03	

- Redis

Tabla 6 – CVE Details Redis

Detalles de vulnerabilidad	Descripción	Puntaje CVSS y fecha de publicación	Fuente
CVE-2019-10193	Se encontró una vulnerabilidad de desbordamiento de búfer de pila en las versiones 3.x de la estructura de datos del hiperloglog de Redis antes de la 3.2.13, 4.x antes de la 4.0.14 y 5.x antes de la 5.0.4. Al corromper un hiperloglog con el comando SETRANGE, un atacante podría hacer que Redis realice incrementos controlados de hasta 12 bytes después del final de un búfer asignado a la pila.	Puntaje CVSS 6.5 Fecha de publicación: 2019-07-11	(CVE, D., *10193*)
CVE-2018-12453	Escriba confusión en la función xgroupCommand en t_stream.c en redis-server en Redis antes de 5.0, lo que permite a atacantes remotos causar denegación de servicio a través de un comando XGROUP en el que la clave no es una secuencia.	Puntaje CVSS 5.0 Fecha de publicación: 2018-06-16	(CVE, D., *12453*)
CVE-2018-12326	El desbordamiento de búfer en redis-cli de Redis antes de 4.0.10 y 5.x antes de 5.0 RC3 permite a un atacante lograr la ejecución del código y escalar a privilegios más altos a través de una línea de comando diseñada. NOTA: No está claro si existen situaciones comunes en las que redis-cli se usa con, por ejemplo, un argumento -h (también conocido como nombre de host) de una fuente que no es de confianza.	Puntaje CVSS 4.6 Fecha de publicación: 2018-06-17	(CVE, D., *12326*)

8.1.3 Vulnerabilidades y contramedidas

Se hizo la recolección de las principales vulnerabilidades y contramedidas que han sido documentadas por parte de investigadores orientado al área de amenazas de seguridad en base de datos NoSQL

Tabla 7 – Vulnerabilidades y Contramedidas

Nombre	Descripción	Contramedidas	Fuente
Autenticación de usuarios inactivos.	No se activó la autenticación de usuario.	Considerar los parámetros que pueden ser útiles para la mayoría de instalaciones y se configuran en el fichero de configuración principal /etc/mongod.conf	(EYMAR SILVA, 2017)
Acceso vía http a la base de datos	Se tiene habilitado el acceso vía HTTP permitiendo manipular datos.	Deshabilitar cualquier acceso vía http tanto a la parte de administración como a la API Rest. http: -enabled: false - RESTInterfaceEnabled: false	(EYMAR SILVA, 2017)
Combinar análisis estático y dinámico	Una de estas soluciones sugeridas se llama AMNESIA. Consta en el uso de una combinación de análisis dinámico para evitar la inyección. Esta herramienta realiza primero el análisis estático de un nivel de presentación escrito por Java. El resultado del análisis estático es a la vez una lista de puntos en el código donde el nivel de presentación envía consultas al DBMS y modelos de consultas legales para esos puntos.	La herramienta realiza análisis dinámico cuando el nivel de presentación envía realmente una consulta al DBMS. Eso compara la consulta del usuario con el modelo de consultas legales generadas durante el análisis estático. Si la consulta del usuario coincide con una consulta legal, la consulta de usuario se pasa al DBMS, de lo contrario, se rechaza la consulta del usuario.	(Jonathan P, 2015)
Inyección NOSQL	No se establecieron los parámetros básicos de seguridad ni se creó usuario para el acceso a la base de datos con privilegios específicos. Se puede ejecutar inyección NOSQL.	Crear usuario administrador con el rol de “root” en la base de datos principal. Crear usuario con el rol de “dwOwner” en la base de datos.	(EYMAR SILVA, 2017)

		Modificar el fichero .config con la configuración y reiniciar el servicio de Mongo	
Conexión abierta desde cualquier ip	Se permite la conexión desde cualquier ip, permitiendo realizar una inyección NoSQL.	Permitir conexiones únicamente desde la ip que se requiera y cambiar puerto por defecto. Ejemplo desde ip local: net: -bindIp: 127.0.0.1 -port: 27019 (el que se requiera)	(EYMAR SILVA, 2017)
Exposición de Datos Sensibles	En su gran mayoría las aplicaciones web no protegen adecuadamente los datos sensibles que administra, pueden ser números de tarjetas de crédito, datos de autenticación entre otros.	No almacenar datos sensibles innecesariamente. Prescindir tan pronto como sea posible. Aplicar algoritmos de cifrado fuertes y estandarizados, así como claves robustas gestionándolas de forma segura. Utilizar módulos criptográficos validados como FIPS 140 del gobierno de Estados Unidos que incluye componentes software y hardware. No guardar contraseñas cuando el navegador lo solicita	(EYMAR SILVA, 2017)
Cabecera XXSS-Protection no está activada	La cabecera X-XSSProtection no se está utilizando, no tiene activo el filtro XSS, el cual puede ser aprovechado para realizar ataques XSS	Una forma de agregar esta cabecera sería adicionando unas líneas de código a los archivos de funciones PHP del tema que se esté usando. Otra posibilidad para habilitar la cabecera en un servidor web como Apache, sería agregando el siguiente código en el fichero .htaccess: Header set X-XSS-Protection "1; mode=block"	(EYMAR SILVA, 2017)

8.1.4 Herramientas de pentesting

Se hizo la recolección de las principales herramientas de penetración o pentest ya que están enfocadas en atacar un sistema informático con la intención de encontrar las debilidades de seguridad y de esta manera poder generar un ataque.

Tabla 8 - Herramientas de pentesting

Nombre	Descripción	Gestor	Fuente
Nessus	Nessus es una herramienta desarrollada para realizar escaneos de seguridad sobre diversos sistemas operativos y aplicaciones. Uno de sus grandes fuertes son los plugins desarrollados para evaluar vulnerabilidades específicas y auditar el nivel de seguridad de los sistemas.	MongoDB, Apache Cassandra y Redis	(Catoira, 2013)
NMAP	Es una herramienta gratuita de código abierto para la exploración de vulnerabilidades y la detección de redes. Los administradores de red utilizan Nmap para identificar qué dispositivos se están ejecutando en sus sistemas, descubrir los hosts disponibles y los servicios que ofrecen, encontrar puertos abiertos y detectar riesgos de seguridad.	MongoDB, Apache Cassandra y Redis	(Marin de la Fuente, 2019)
NIKTO	Nikto es una herramienta de escaneo de servidores web que se encarga de efectuar diferentes tipos de actividades tales como, detección de malas configuraciones y vulnerabilidades en el servidor objetivo, detección de ficheros en instalaciones por defecto, listado de la estructura del servidor, versiones y fechas de actualizaciones de servidores, tests de vulnerabilidades XSS, ataques de fuerza bruta por diccionario, reportes en formatos txt, csv, html, etc.	MongoDB, Apache Cassandra y Redis	(Adastra, 2011)
LEGION	Legion una herramienta de prueba de penetración de red fácil de usar, súper extensible y semi automática que ayuda en el descubrimiento, reconocimiento y explotación de sistemas de información	MongoDB, Apache Cassandra y Redis	(Underc0de, 2019)

8.2 Fase 2. Elaboración del plan de pruebas.

Se planteó un procedimiento para la protección de ataques por inyección SQL en bases de datos no relacionales mediante el estudio y la búsqueda de técnicas aplicadas a partir de un escenario de pruebas que se dividen en una serie de fases las cuales son:

Tabla 9 – Plan de Pruebas

Fase	Descripción	Herramientas
Recopilación de información	En esta fase se establecen los objetivos y se busca información sobre los gestores de bases de datos, lenguajes de programación con los que interactúan y configuración por defecto para poder entender como atacar.	- Nmap
Análisis de vulnerabilidades	En esta fase se usan los datos obtenidos en la fase anterior para seleccionar las vulnerabilidades que se pueden ejecutar.	- Nessus - Nikto - Legion
Explotación	Una vez identificadas las vulnerabilidades se deduce cuáles de ellas pueden ser atacadas y de qué forma.	- Metasploit
Reporte	Finalmente se realiza un registro sobre los resultados de los ataques ejecutados en la fase de explotación y se proponen soluciones que puedan evitar los ataques.	

Fuente El autor

8.2.1 Recopilación de información

Es aconsejable conseguir información asociada con el nombre del host, la dirección IP puede conseguirse con algunos métodos como por ejemplo preguntas al DNS, realizar búsquedas en InterNIC (WHOIS) y haciendo sniffing a la red. Para la recolección de información se pueden tener en cuenta las siguientes actividades:

a. Descubrimiento en la red: la intención es ver los equipos activos en la red.

b. Identificación de puertos y servicios de red: se utiliza un analizador de puertos, con el fin de detectar los puertos abiertos de un equipo, así es posible identificar el tipo de sistema operativo.

8.2.2 Análisis de Vulnerabilidades

Se fundamenta en la comparación de los servicios, aplicaciones y sistemas operativos de los hosts escaneados enfrentándose con una base de datos de vulnerabilidades.

a. Apoya a la identificación de versiones obsoletas de software, parches no instalados, configuraciones erradas y valida el cumplimiento de las políticas de seguridad de la organización.

b. Con el conocimiento de las vulnerabilidades de los sistemas de la organización, es posible concretar la estrategia para realizar la prueba de penetración a los equipos de la red.

8.2.3 Explotación

Una vez se tienen identificadas las vulnerabilidades, el pentester o equipo de pruebas trata de comprender más sobre la estructura y comportamiento de la red con el fin de explotar las vulnerabilidades establecidas. Algunos exploits admiten elevar privilegios en el sistema o en la red para acceder a recursos adicionales. Es recomendable desplazarse por el sistema en búsqueda de más información, esto permitirá efectuar un descubrimiento adicional en busca de más vulnerabilidades.
Fase de explotación.

Se puede categorizar las vulnerabilidades explotadas en esta investigación de la siguiente manera:

a. Errores de configuración.
b. Debilidades en la validación de posibles entradas en una aplicación.
c. Privilegios accedidos. Es posible aprovechar los privilegios superiores durante el tiempo que un programa o proceso está todavía en modo de ejecución.
d. Inadecuada asignación de permisos a archivos y directorios.

8.2.4 Reporte

Los datos al ser recopilados deben ser analizados detalladamente, con el fin de proporcionar un informe sobre los resultados obtenidos y comparados entre los tres gestores de bases de datos NoSQL y un informe donde se propongan contramedidas.

8.3 Fase 3. Implementación del prototipo y pruebas.

8.3.1 Prototipo

Inicialmente es necesario preparar un entorno en donde se implementa un prototipo. Esto requiere una configuración completa desde la instalación de sistemas operativos en máquinas virtuales con un proceso de instalación de gestores de

bases de datos y lenguajes de programación como se describe en Véase anexo 1 hasta el desarrollo de dicho prototipo el cual es descrito en el véase anexo 2.

Específicamente las herramientas utilizadas que se describen a detalle en los anexos son: VirtualBox, MongoDB, Cassandra, Redis y PHP. Las máquinas virtuales utilizan los sistemas operativos de Kali desde donde se realizan los ataques y Ubuntu 20.04 en donde se ejecutan los prototipos. La ventaja de utilizar máquinas virtuales es que se generan puntos de guardado con configuraciones independientes para que un gestor de bases de datos no genere ningún conflicto en sus operaciones con otro gestor.

En el siguiente aparatado se implementa el plan de pruebas sobre cada prototipo.

8.3.2 Recopilación de información sobre Prototipo

Inicialmente se ejecutan los tres prototipos con la funcionalidad de PHP que asigna un servidor para máquinas remotas a un puerto determinado. El siguiente comando permite asignar un puerto donde se ejecuta el prototipo en este caso es el puerto 8000 y la dirección 0.0.0.0 permite que se pueda acceder con la dirección ip desde otra máquina. Este comando se debe ejecutar desde el directorio del proyecto.

- $ php -S 0.0.0.0:8000

Nmap sobre MongoDB

Se utiliza la máquina atacante de Kali para realizar el análisis de los puertos de la máquina víctima por medio de las herramientas Nmap, Nikto y Legion.

Ilustración 9 – Escaneo de Puertos – MongoDB - Nmap

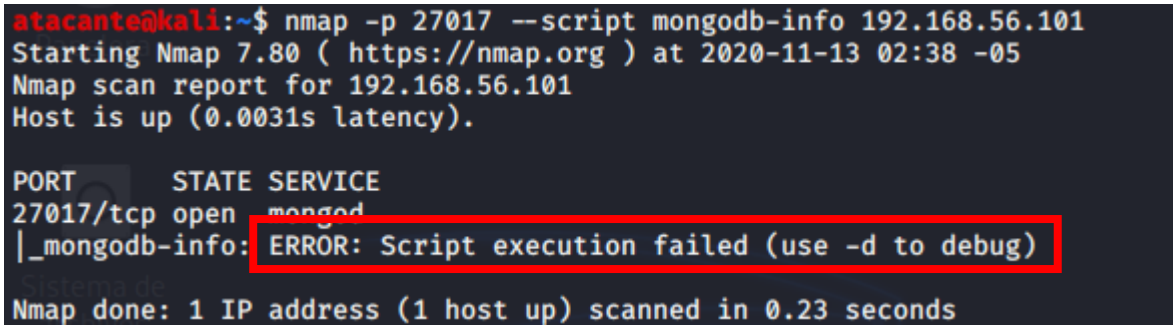
```
atacante@kali:~$ nmap -sV 192.168.56.101 -p-
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-10 17:41 -05
Nmap scan report for 192.168.56.101
Host is up (0.011s latency).
Not shown: 65532 filtered ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.41 ((Ubuntu))
8000/tcp  open  http    PHP cli server 5.5 or later (PHP 7.4.11)
27017/tcp open  mongodb MongoDB 3.6.8

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
 .
Nmap done: 1 IP address (1 host up) scanned in 140.82 seconds
atacante@kali:~$
```

Fuente El autor

Mediante Nmap se filtran todos los puertos abiertos de la máquina víctima, en el puerto 80 se obtiene la versión de apache, en el puerto 8000 se ejecuta el prototipo con la versión 7.4.11 de PHP y finalmente sobre el puerto de MongoDB se obtiene

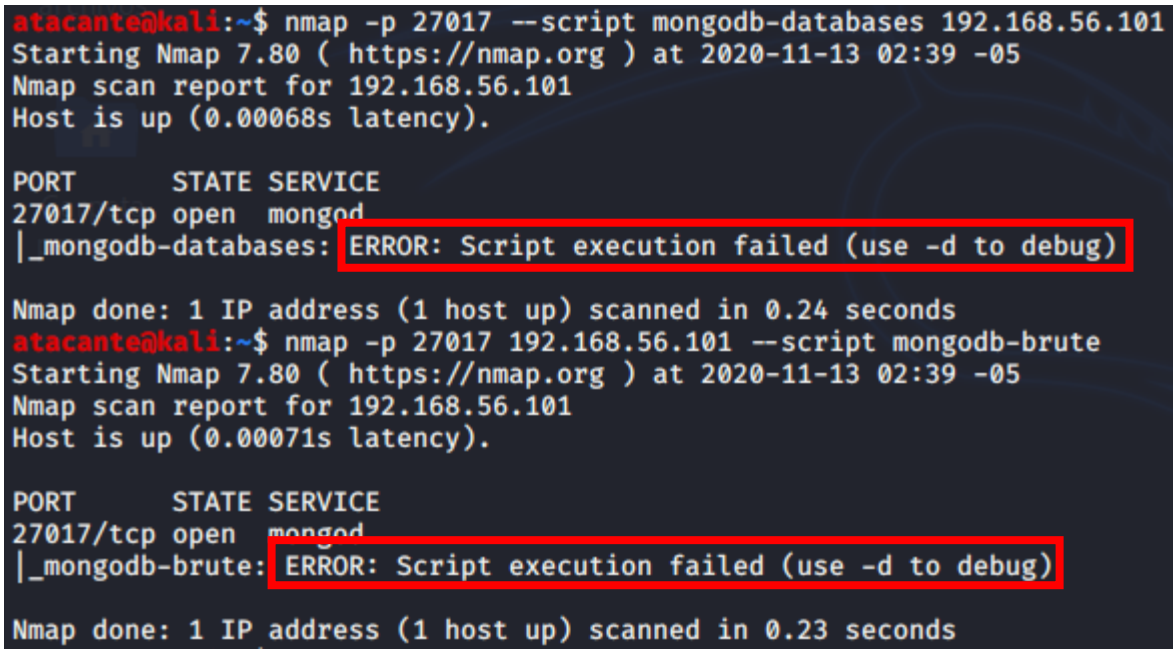
la versión que es 3.6.8. está activo y abierto. También se utilizó el complemento -sV mediante el cual se obtienen las versiones de los servicios que pertenecen a los puertos y el complemento -p- es usado para realizar un análisis sobre todos los puertos abiertos.

Ilustración 10 - Puertos abiertos - MongoDB



Fuente El autor

Ilustración 11 - Puertos abiertos - MongoDB



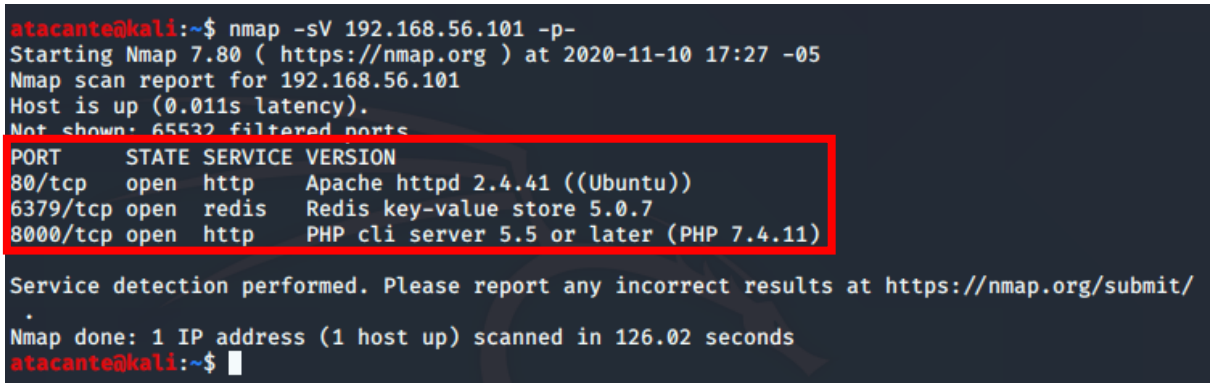
Fuente El autor

Nmap también provee herramientas enfocadas a un objetivo específico, para este caso los scripts para obtener información sobre la base de datos incluidos los registros y contraseñas para usuarios no se ejecutan así que no generan resultados.

Nmap sobre Redis

Se utiliza la máquina atacante de Kali para realizar el análisis de los puertos de la máquina víctima por medio de las herramientas Nmap, Nikto y Legion.

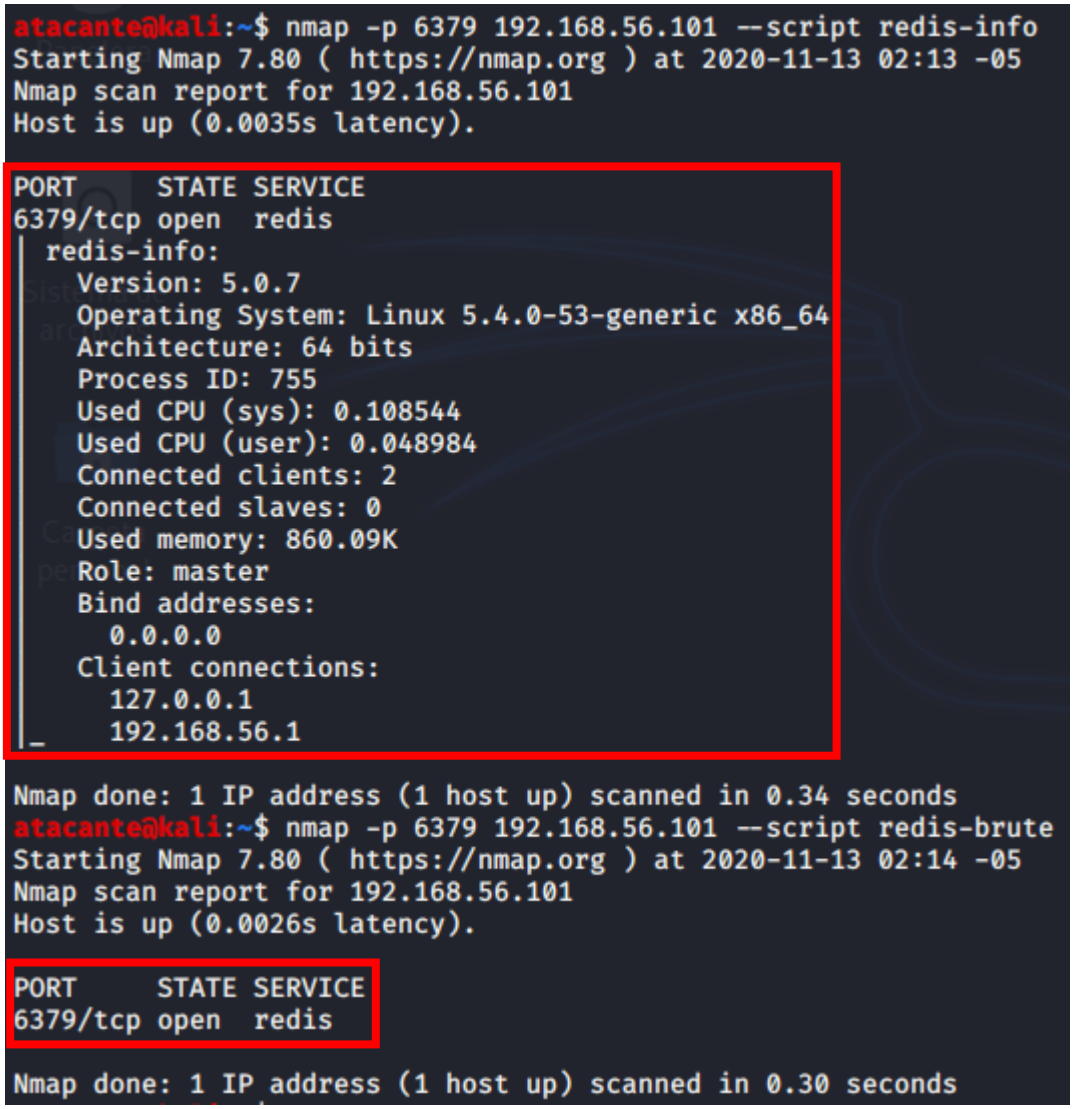
Ilustración 12 - Escaneo de Puertos – Redis - Nmap



Fuente El autor

Mediante Nmap se filtran todos los puertos abiertos de la máquina víctima, en el puerto 80 se obtiene la versión de apache, en el puerto 8000 se ejecuta el prototipo con la versión 7.4.11 de PHP y finalmente sobre el puerto de Redis se obtiene la versión que es 5.0.7. está activo y abierto. También se utilizó el complemento -sV mediante el cual se obtienen las versiones de los servicios que pertenecen a los puertos y el complemento -p- es usado para realizar un análisis sobre todos los puertos abiertos.

Ilustración 13 - Puertos abiertos Redis



Fuente El autor

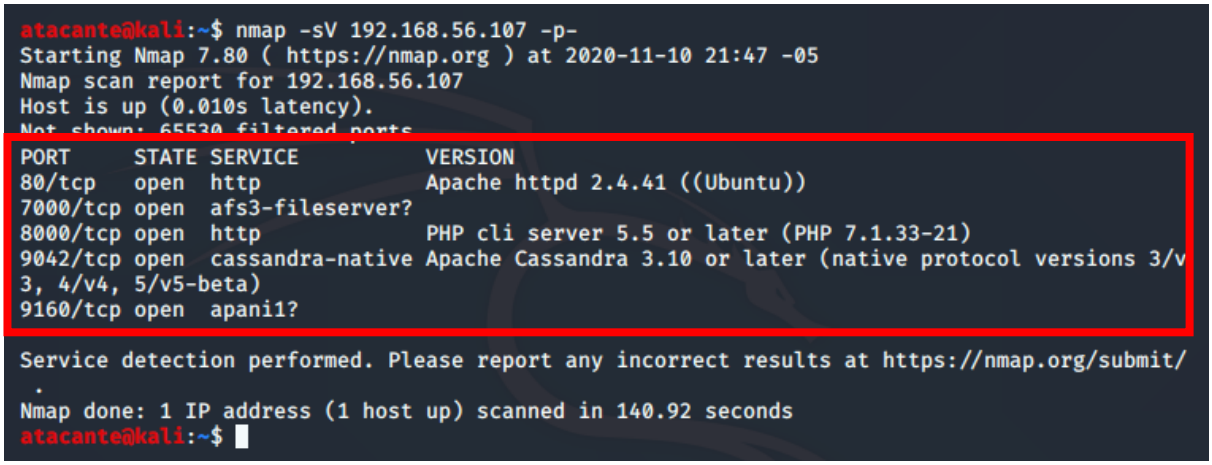
Nmap también provee herramientas enfocadas a un objetivo específico, para este caso el script redis-info que solo se puede ejecutar cuando se encuentra abierto el puerto 6379 donde se ejecuta la consola de Redis e intenta obtener información básica y direcciones de conexiones.

El segundo script es redis-brute el cual realiza un ataque de fuerza bruta en la base de datos de Redis. El resultado obtenido informa que la configuración en la base de datos se realizó por defecto así que no hay usuario ni contraseña.

Nmap sobre Apache Cassandra

Se utiliza la máquina atacante de Kali para realizar el análisis de los puertos de la máquina víctima por medio de las herramientas Nmap, Nikto y Legion.

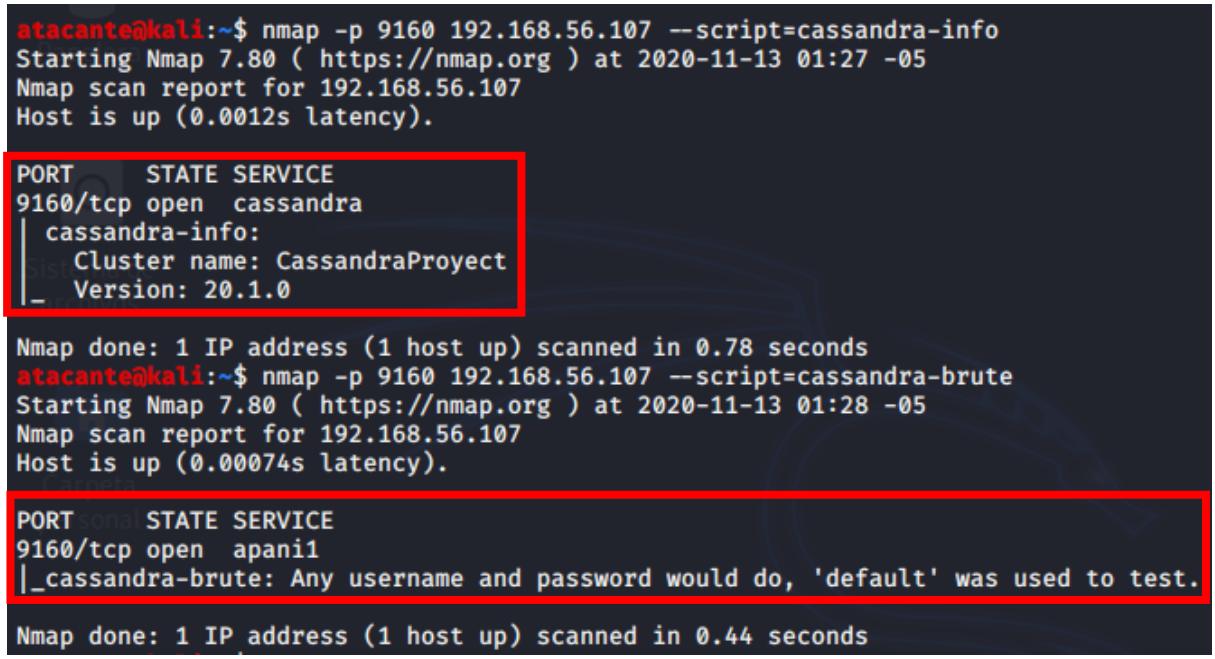
Ilustración 14 - Escaneo de Puertos – Cassandra - Nmap



Fuente El autor

Mediante Nmap se filtran todos los puertos abiertos de la máquina víctima, en el puerto 80 se obtiene la versión de apache, en el puerto 8000 se ejecuta el prototipo con la versión 7.1.33 de PHP (a diferencia de los otros dos gestores no es compatible con la versión más reciente) y finalmente sobre el puerto de la consola de comandos Cassandra se obtiene que la versión es más reciente a la 3.10. lo cual es correcto debido a que en la máquina se puede ver que es la 3.11.8. Existen dos puertos adicionales, el puerto 7000 el cual proporciona comunicación entre clústeres y el puerto 9160 que es donde se encuentra la información del clúster. También se utilizó el complemento -sV mediante el cual se obtienen las versiones de los servicios que pertenecen a los puertos y el complemento -p- es usado para realizar un análisis sobre todos los puertos abiertos.

Ilustración 15 - Puertos abiertos Cassandra



Fuente El autor

Nmap también provee herramientas enfocadas a un objetivo específico, para este caso el script cassandra-info que solo se puede ejecutar cuando se encuentra abierto el puerto 9160 “este puerto usa el Protocolo de Control de Transmisión que, en otras palabras, es un protocolo orientado en la conexión, garantizando la entrega de paquetes de datos” de esta manera intenta obtener información básica del estado del servidor de una base de datos de Cassandra. El script muestra el nombre del clúster junto a su versión para este caso 20.1.0.

El segundo script es cassandra-brute el cual realiza un ataque de fuerza bruta en la base de datos de Cassandra. El resultado obtenido informa que la configuración en la base de datos se realizó por defecto así que no hay usuario ni contraseña.

8.3.3 Análisis de Vulnerabilidades sobre Prototipo

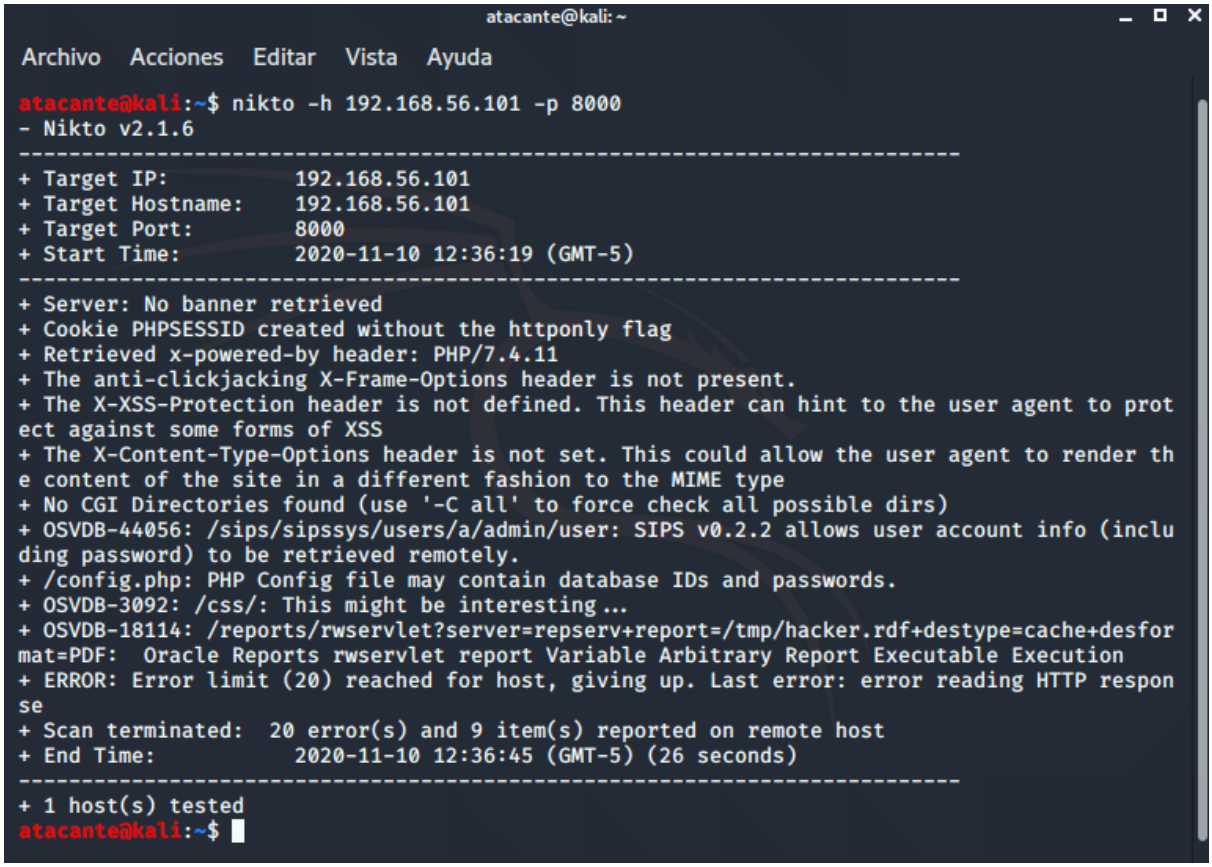
Recopilación de información (Nikto)

Mediante la herramienta de software gratuito Nikto se puede realizar escaneos de seguridad a los servidores del prototipo, con el fin de buscar posibles vulnerabilidades.

Nikto sobre MongoDB

Se utiliza la máquina atacante de Kali para realizar el análisis de los puertos de la máquina víctima por medio de las herramientas Nessus, Nikto y Legion.

Ilustración 16 - Escaneo de Puerto 8000 – MongoDB - Nikto



Fuente El autor

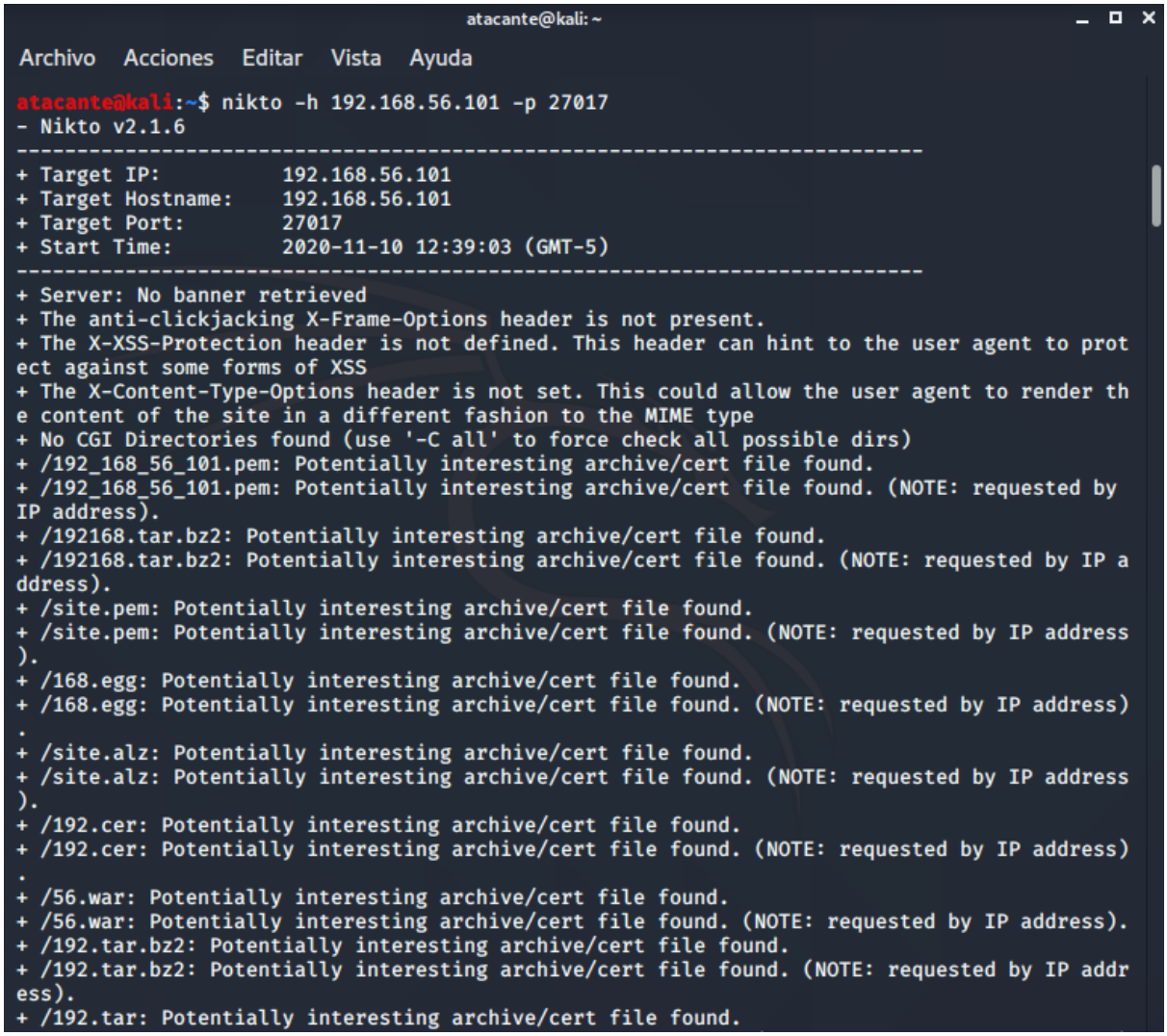
Los resultados de este escaneo no revelan ningún banner de respuesta, algunos temas relacionados con las cabeceras de respuesta devueltas, tecnologías relacionadas, y una gran cantidad de potenciales temas de seguridad o vulnerabilidades identificadas Como:

- El encabezado X-XSS-Protection no está definido. Este encabezado puede sugerirle al agente de usuario que se proteja contra algunas formas de XSS

- The anti-clickjacking X-Frame-Options header is not present
    - El servidor no devolvió un encabezado X-Frame-Options, lo que significa que este sitio web podría estar en riesgo de un ataque de secuestro de clics. El encabezado de respuesta HTTP X-Frame-

Options se puede utilizar para indicar si se debe permitir o no que un explorador represente una página dentro de un marco o frame.

Ilustración 17 - Escaneo de Puerto 27017 – MongoDB - Nikto



Fuente El autor

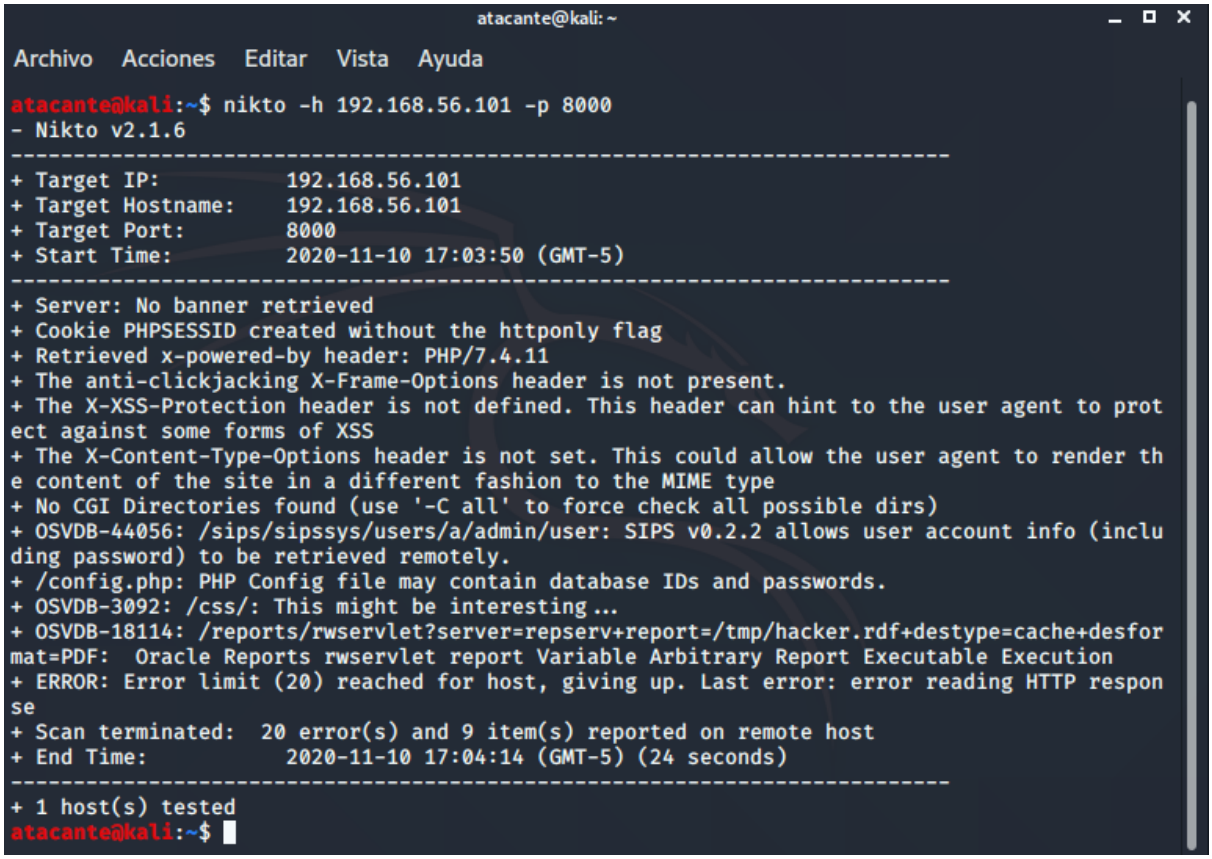
Los resultados de este escaneo no revelan ningún banner de respuesta, algunos temas relacionados con las cabeceras de respuesta devueltas, y una gran cantidad de potenciales temas de seguridad o vulnerabilidades identificadas.

Todos estos hallazgos deben ser verificados manualmente, pues podrían ser falsos positivos generadas por la herramienta Nikto.

Nikto sobre Redis

Se utiliza la máquina atacante de Kali para realizar el análisis de los puertos de la máquina víctima por medio de las herramientas Nessus, Nikto y Legion.

Ilustración 18 - Escaneo de Puerto 8000 – Redis - Nikto



Fuente El autor

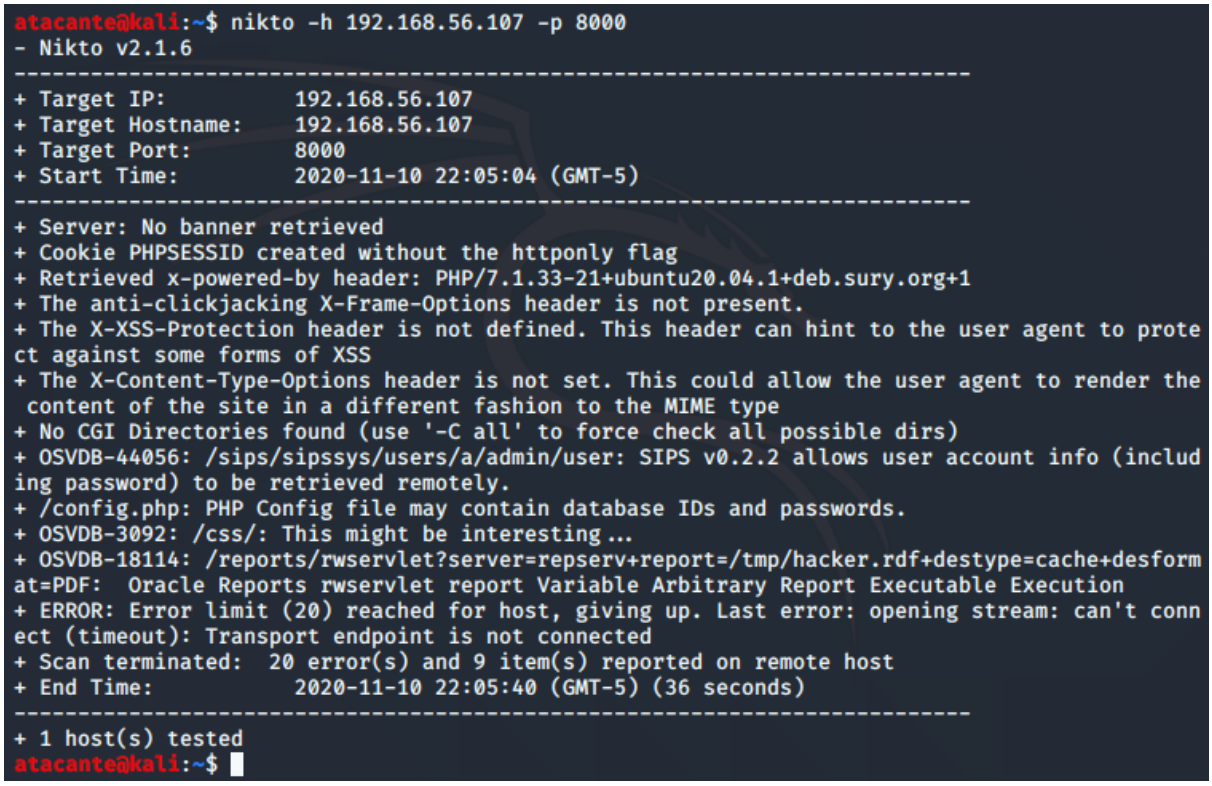
Los resultados de este escaneo no revelan ningún banner de respuesta, algunos temas relacionados con las cabeceras de respuesta devueltas, tecnologías relacionadas, y temas relacionados a potenciales vulnerabilidades como:

- Cookie PHPSESSID created without the httponly flag
  - Si el atributo HttpOnly está configurado en una cookie, el valor de la cookie no se puede leer ni configurar mediante JavaScript del lado del cliente. Esta medida hace que ciertos ataques del lado del cliente, como las secuencias de comandos entre sitios, sean un poco más difíciles de explotar al evitar que capturen trivialmente el valor de la cookie a través de una secuencia de comandos inyectada.
  - HttpOnly Flag: El uso de HttpOnly en Set-Cookie ayuda a mitigar el riesgo más común de una Ataque XSS.

Nikto sobre Apache Cassandra

Se utiliza la máquina atacante de Kali para realizar el análisis de los puertos de la máquina víctima por medio de las herramientas Nessus, Nikto y Legion.

Ilustración 19 - Escaneo de Puerto 8000 – Cassandra - Nikto



Fuente El autor

Los resultados de este escaneo no revelan ningún banner de respuesta, algunos temas relacionados con las cabeceras de respuesta devueltas, tecnologías relacionadas, y temas relacionados a potenciales vulnerabilidades como:

- Retrieved x-powered-by header: PHP/7.1 .33-24+ ubuntu20.04.1+deb.sury.org+1
  - Errores en la instalación de PHP en Ubuntu
- Cookie PHPSESSID created without the httponly flag
  - Si el atributo HttpOnly está configurado en una cookie, el valor de la cookie no se puede leer ni configurar mediante JavaScript del lado del cliente. Esta medida hace que ciertos ataques del lado del cliente, como las secuencias de comandos entre sitios, sean un poco más difíciles de explotar al evitar que capturen trivialmente el valor de la cookie a través de una secuencia de comandos inyectada.

- HttpOnly Flag: El uso de HttpOnly en Set-Cookie ayuda a mitigar el riesgo más común de una Ataque XSS.

Recopilación de información (Nessus)

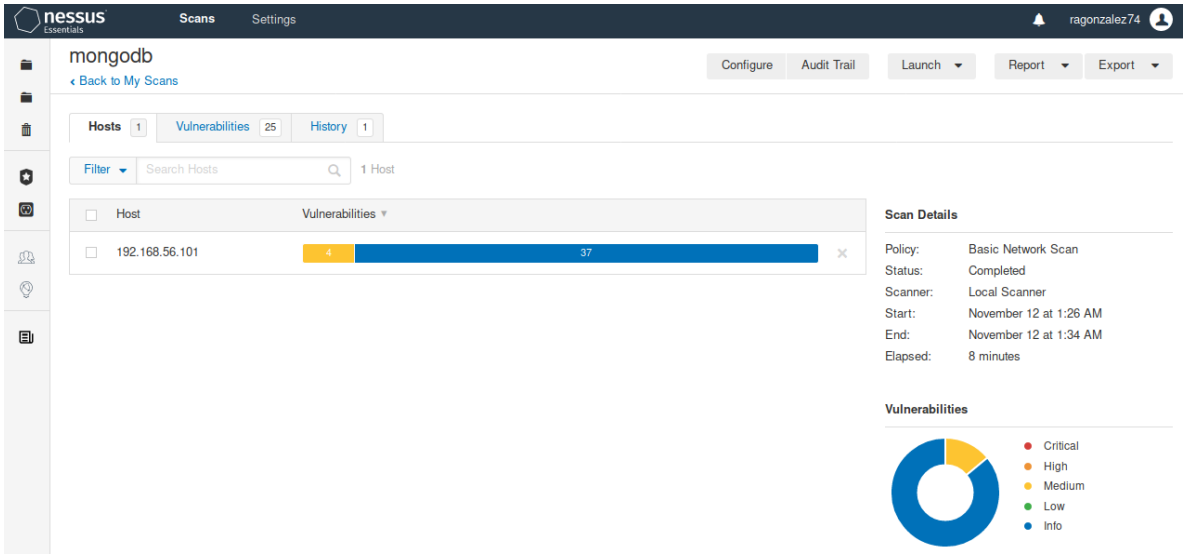
Se debe contar con esta herramienta de escaneo de vulnerabilidades con el fin de encontrar toda la información disponible sobre amenazas o vulnerabilidades que el prototipo podría presentar por medio de la ventaja que proveen sus búsquedas tan específicas. El escaneo se realiza en cada una de las máquinas en donde están instalados los gestores de bases de datos NoSQL.

Al finalizar el escaneo de seguridad se obtiene una gran cantidad de resultados con las posibles amenazas que pueden tener los gestores de bases de datos NoSQL, se hizo el análisis de las amenazas que tuvieran una puntuación alta tanto en su factor de riesgo como su puntuación base de CVSS y se obtuvo lo siguiente:

Nessus sobre MongoDB

Los siguientes datos hacen parte del escaneo de seguridad y fueron escogidos para su análisis según su factor de importancia. Al ver los resultados se puede llegar a comprender la situación de seguridad de la base de datos de MongoDB, con la ayuda de los indicadores de color y las opciones de visualización, Nessus nos brindan una visión de los riesgos potenciales.

Ilustración 20 - Escaneo Vulnerabilidad – MongoDB - Nessus



Fuente El autor

Tabla 10 – Vulnerabilidad 65702 – MongoDB - Nessus

Nombre de la vulnerabilidad	Repositorio Git servido por servidor web				# Vulnerabilidad	65702
Sinopsis	El servidor web remoto puede revelar información debido a una debilidad de configuración.				Factor de riesgo	Medio
Descripción	El servidor web del host remoto permite el acceso de lectura a un repositorio Git. Este posible defecto se puede utilizar para descargar contenido del servidor Web que de otro modo podría ser privado.					
Contramedida	Compruebe que los repositorios de Git enumerados se sirven intencionadamente.					
Puntuación base de CVSS	Vector de ataque	Complejidad de ataque	Interacción del usuario	Confidencialidad	Integridad	Disponibilidad
5.0	AV:N	AC:L	Au:N	C:P	I:N	A:N

Fuente El autor

Tabla 11 – Vulnerabilidad 81777 – MongoDB - Nessus

Nombre de la vulnerabilidad	Servicio MongoDB sin detección de autenticación				# Vulnerabilidad	81777
Sinopsis	El host remoto está ejecutando un sistema de base de datos que no tiene habilitada la autenticación.				Factor de riesgo	Medio
Descripción	MongoDB, un sistema de base de datos orientado a documentos, está escuchando en el puerto remoto y está configurado para permitir conexiones sin ninguna autenticación. Por lo tanto, un atacante remoto puede conectarse al sistema de base de datos para crear, leer, actualizar y eliminar documentos, colecciones y bases de datos.					
Contramedida	Habilite la autenticación o restrinja el acceso al servicio MongoDB.					
Puntuación base de CVSS	Vector de ataque	Complejidad de ataque	Interacción del usuario	Confidencialidad	Integridad	Disponibilidad
6.4	AV:N	AC:L	Au:N	C:P	I:P	A:N

Fuente El autor

Tabla 12 – Vulnerabilidad 85582 – MongoDB - Nessus

Nombre de la vulnerabilidad	Aplicación web potencialmente vulnerable al secuestro de clics				# Vulnerabilidad	85582
Sinopsis	Es posible que el servidor web remoto no pueda mitigar una clase de vulnerabilidades de aplicaciones web.				Factor de riesgo	Medio
Descripción	El servidor web remoto no establece un encabezado de respuesta X-Frame-Options o un encabezado de respuesta. Esto podría exponer potencialmente el sitio a un ataque de corrección de clics o de interfaz de usuario, en el que un atacante puede					

	engañar a un usuario para que haga clic en un área de la página vulnerable que es diferente de lo que el usuario percibe que es la página.					
Contramedida	Devuelve el encabezado HTTP X-Frame-Options o Content-Security-Policy (con la directiva 'frame-ancestors') con la respuesta de la página. Esto evita que otro sitio represente riesgos en el contenido de la página cuando se utiliza el marco o las etiquetas HTML tags de iframe.					
Puntuación base de CVSS	Vector de ataque	Complejidad de ataque	Interacción del usuario	Confidencialidad	Integridad	Disponibilidad
4.3	AV:N	AC:M	Au:N	C:N	I:P	A:N

Fuente El autor

Tabla 13 – Vulnerabilidad 65702 – MongoDB - Nessus

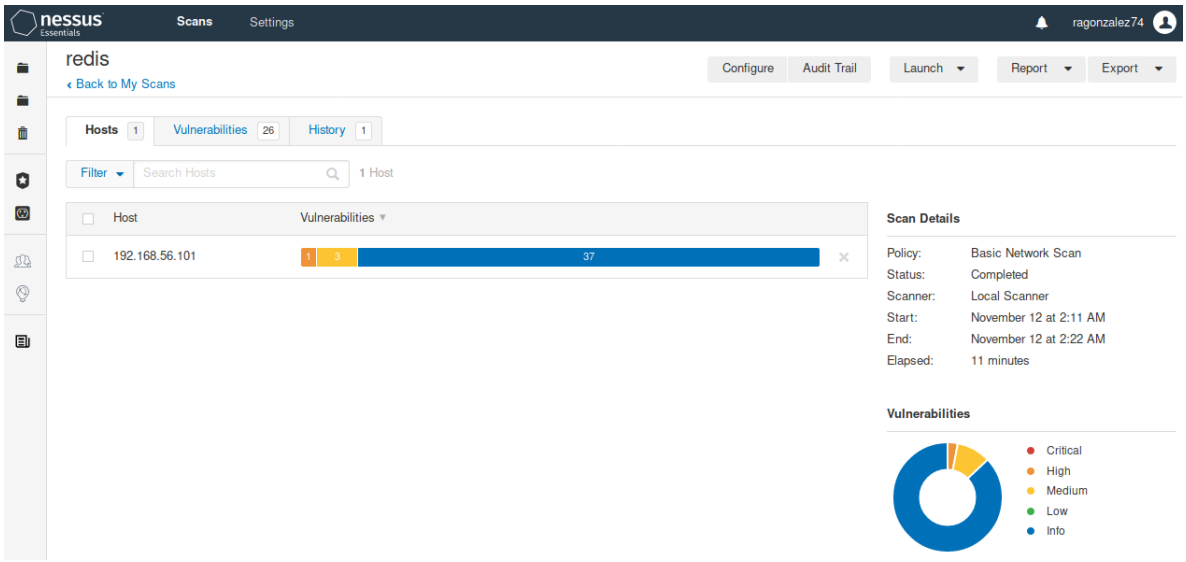
Nombre de la vulnerabilidad	Detección de mDNS (red remota)			# Vulnerabilidad	12218
Sinopsis	Es posible obtener información sobre el host remoto.			Factor de riesgo	Medio
Descripción	El servicio remoto comprende el protocolo Bonjour (también conocido como ZeroConf o mDNS), que permite a cualquier persona descubrir información del host remoto, como su tipo de sistema operativo y la versión exacta, su nombre de host, y la lista de servicios que está ejecutando. Este plugin intenta descubrir el protocolo mDNS utilizado por el host y debe verificar que el protocolo no esté en el segmento de red en el que reside Nessus.					
Contramedida	Filtre el tráfico entrante al puerto UDP 5353, si lo desea.					
Puntuación base de CVSS	Vector de ataque	Complejidad de ataque	Interacción del usuario	Confidencialidad	Integridad	Disponibilidad
5.0	AV:N	AC:L	Au:N	C:P	I:N	A:N

Fuente El autor

Nessus sobre Redis

Los siguientes datos hacen parte del escaneo de seguridad y fueron escogidos para su análisis según su factor de importancia. Al ver los resultados se puede llegar a comprender la situación de seguridad de la base de datos de Redis, con la ayuda de los indicadores de color y las opciones de visualización, Nessus nos brindan una visión de los riesgos potenciales en sus activos.

Ilustración 21 - Escaneo Vulnerabilidad – Redis - Nessus



Fuente El autor

Tabla 14 – Vulnerabilidad 100643 – Redis - Nessus

Nombre de la vulnerabilidad	Redis Server Desprotegido por autenticación de contraseña			# Vulnerabilidad	100634
Sinopsis	Un servidor de Redis no está protegido por autenticación por contraseña.			Factor de riesgo	Alto
Descripción	El servidor de Redis que se ejecuta en el host remoto no está protegido por autenticación por contraseña. Un atacante remoto puede explotar esto para obtener acceso no autorizado al servidor.				
Contramedida	Habilite la directiva 'requirepass' en el archivo de configuración redis.conf.				
Puntuación base de CVSS	Vector de ataque	Complejidad de ataque	Interacción del usuario	Confidencialidad	Integridad	Disponibilidad
7.5	AV:N	AC:L	Au:N	C:P	I:P	A:P

Fuente El autor

Tabla 15 – Vulnerabilidad 65702 – Redis - Nessus

Nombre de la vulnerabilidad	Repositorio Git servido por servidor web				# Vulnerabilidad	65702
Sinopsis	El servidor web remoto puede revelar información debido a una debilidad de configuración.				Factor de riesgo	Medio
Descripción	El servidor web del host remoto permite el acceso de lectura a un repositorio Git. Este posible defecto se puede utilizar para descargar contenido del servidor Web que de otro modo podría ser privado.					
Contramedida	Compruebe que los repositorios de Git sirven intencionadamente.					
Puntuación base de CVSS	Vector de ataque	Complejidad de ataque	Interacción del usuario	Confidencialidad	Integridad	Disponibilidad
5.0	AV:N	AC:L	Au:N	C:P	I:N	A:N

Fuente El autor

Tabla 16 – Vulnerabilidad 85582 – Redis - Nessus

Nombre de la vulnerabilidad	Aplicación web potencialmente vulnerable al secuestro de clics				# Vulnerabilidad	85582
Sinopsis	Es posible que el servidor web remoto no pueda mitigar una clase de vulnerabilidades de aplicaciones web.				Factor de riesgo	Medio
Descripción	El servidor web remoto no establece un encabezado de respuesta X-Frame-Options o un encabezado de respuesta. Esto podría exponer potencialmente el sitio a un ataque de corrección de clics o de interfaz de usuario, en el que un atacante puede engañar a un usuario para que haga clic en un área de la página vulnerable que es diferente de lo que el usuario percibe que es la página.					
Contramedida	Devuelve el encabezado HTTP X-Frame-Options o Content-Security-Policy (con la directiva 'frame-ancestors') con la respuesta de la página. Esto evita que otro sitio represente riesgos en el contenido de la página cuando se utiliza el marco o las etiquetas HTML tags de iframe.					
Puntuación base de CVSS	Vector de ataque	Complejidad de ataque	Interacción del usuario	Confidencialidad	Integridad	Disponibilidad
4.3	AV:N	AC:M	Au:N	C:N	I:P	A:N

Fuente El autor

Tabla 17 – Vulnerabilidad 12218 – Redis - Nessus

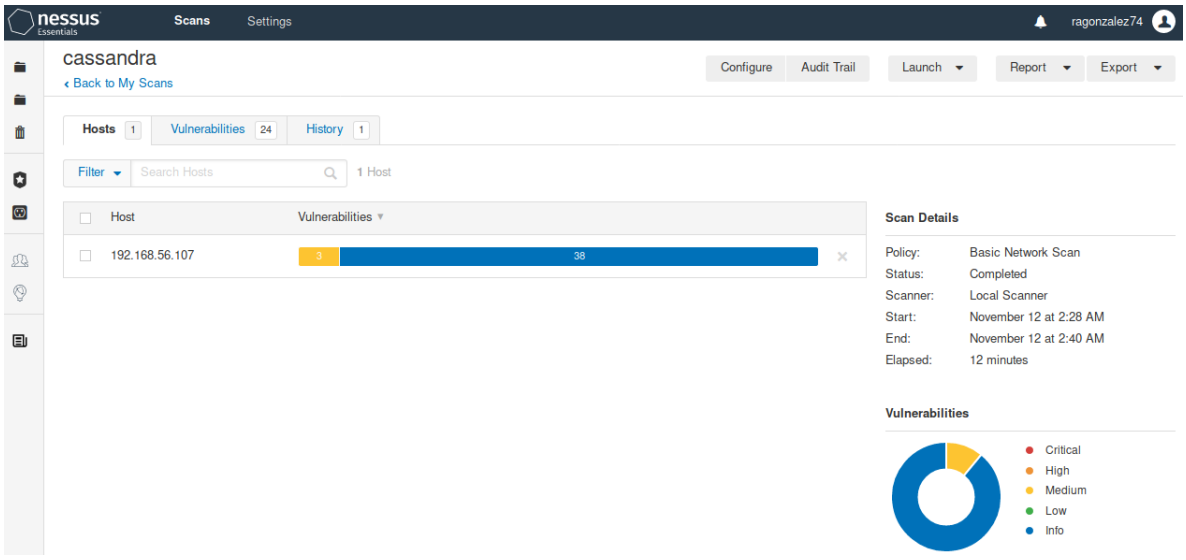
Nombre de la vulnerabilidad	Detección de mDNS (red remota)				# Vulnerabilidad	12218
Sinopsis	Es posible obtener información sobre el host remoto.				Factor de riesgo	Medio
Descripción	El servicio remoto comprende el protocolo Bonjour (también conocido como ZeroConf o mDNS), que permite a cualquier persona descubrir información del host remoto, como su tipo de sistema operativo y la versión exacta, su nombre de host, y la lista de servicios que está ejecutando. Este plugin intenta descubrir mDNS utilizado por hosts que no están en el segmento de red en el que reside Nessus.					
Contramedida	Filtre el tráfico entrante al puerto UDP 5353, si lo desea.					
Puntuación base de CVSS	Vector de ataque	Complejidad de ataque	Interacción del usuario	Confidencialidad	Integridad	Disponibilidad
5.0	AV:N	AC:L	Au:N	C:P	I:N	A:N

Fuente El autor

Nessus sobre Apache Cassandra

Los siguientes datos hacen parte del escaneo de seguridad y fueron escogidos para su análisis según su factor de importancia. Al ver los resultados se puede llegar a comprender la situación de seguridad de la base de datos de Apache Cassandra, con la ayuda de los indicadores de color y las opciones de visualización, Nessus nos brindan una visión de los riesgos potenciales en sus activos.

Ilustración 22 - Escaneo Vulnerabilidad – Cassandra - Nessus



Fuente El autor

Tabla 18 – Vulnerabilidad 65702 – Cassandra - Nessus

Nombre de la vulnerabilidad	Repositorio Git servido por servidor web				# Vulnerabilidad	65702
Sinopsis	El servidor web remoto puede revelar información debido a una debilidad de configuración.				Factor de riesgo	Medio
Descripción	El servidor web del host remoto permite el acceso de lectura a un repositorio Git. Este posible defecto se puede utilizar para descargar contenido del servidor Web que de otro modo podría ser privado.					
Contramedida	Compruebe que los repositorios de Git enumerados se sirven intencionadamente.					
Puntuación base de CVSS	Vector de ataque	Complejidad de ataque	Interacción del usuario	Confidencialidad	Integridad	Disponibilidad
5.0	AV:N	AC:L	Au:N	C:P	I:N	A:N

Fuente El autor

Tabla 19 – Vulnerabilidad 85582 – Cassandra - Nessus

Nombre de la vulnerabilidad	Aplicación web potencialmente vulnerable al secuestro de clics				# Vulnerabilidad	85582
Sinopsis	Es posible que el servidor web remoto no pueda mitigar una clase de vulnerabilidades de aplicaciones web.				Factor de riesgo	Medio
Descripción	El servidor web remoto no establece un encabezado de respuesta X-Frame-Options o un encabezado de respuesta. Esto podría exponer potencialmente el sitio a un ataque de corrección de clics o de interfaz de usuario, en el que un atacante puede engañar a un usuario para que haga clic en un área de la página vulnerable que es diferente de lo que el usuario percibe que es la página.					
Contramedida	Devuelve el encabezado HTTP X-Frame-Options o Content-Security-Policy (con la directiva 'frame-ancestors') con la respuesta de la página. Esto evita que otro sitio represente riesgos en el contenido de la página cuando se utiliza el marco o las etiquetas HTML tags de iframe.					
Puntuación base de CVSS	Vector de ataque	Complejidad de ataque	Interacción del usuario	Confidencialidad	Integridad	Disponibilidad
4.3	AV:N	AC:M	Au:N	C:N	I:P	A:N

Fuente El autor

Tabla 20 – Vulnerabilidad 12218 – Cassandra - Nessus

Nombre de la vulnerabilidad	Detección de mDNS (red remota)				# Vulnerabilidad	12218
Sinopsis	Es posible obtener información sobre el host remoto.				Factor de riesgo	Medio
Descripción	El servicio remoto comprende el protocolo Bonjour (también conocido como ZeroConf o mDNS), que permite a cualquier persona descubrir información del host remoto, como su tipo de sistema operativo y la versión exacta, su nombre de host, y la lista de servicios que está ejecutando. Este plugin intenta descubrir mDNS utilizado por hosts que no están en el segmento de red en el que reside Nessus.					
Contramedida	Filtre el tráfico entrante al puerto UDP 5353, si lo desea.					
Puntuación base de CVSS	Vector de ataque	Complejidad de ataque	Interacción del usuario	Confidencialidad	Integridad	Disponibilidad
5.0	AV:N	AC:L	Au:N	C:P	I:N	A:N

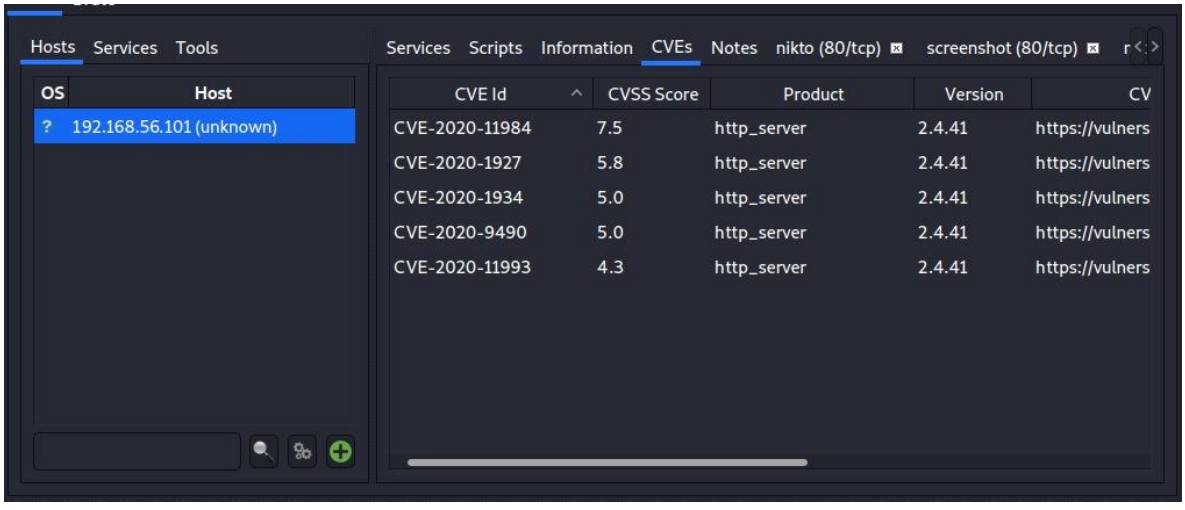
Fuente El autor

Recopilación de información (Legion)

Legion es una herramienta que permite realizar diversas pruebas de penetración que ayuda en el descubrimiento, reconocimiento y explotación de sistemas de información.

Legion sobre MongoDB

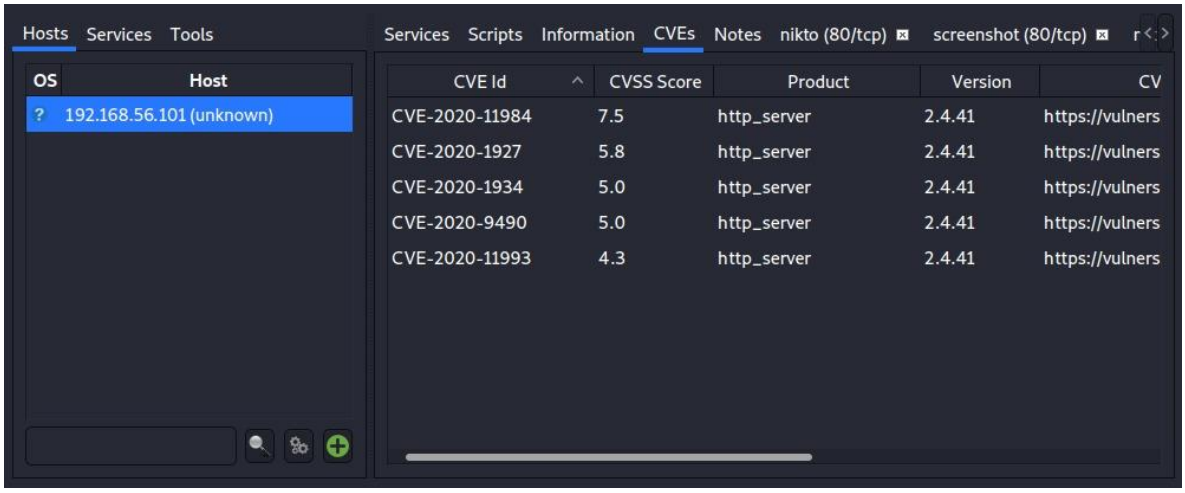
Ilustración 23 - Escaneo de vulnerabilidad – MongoDB - Legion



Fuente El autor

Legion sobre Redis

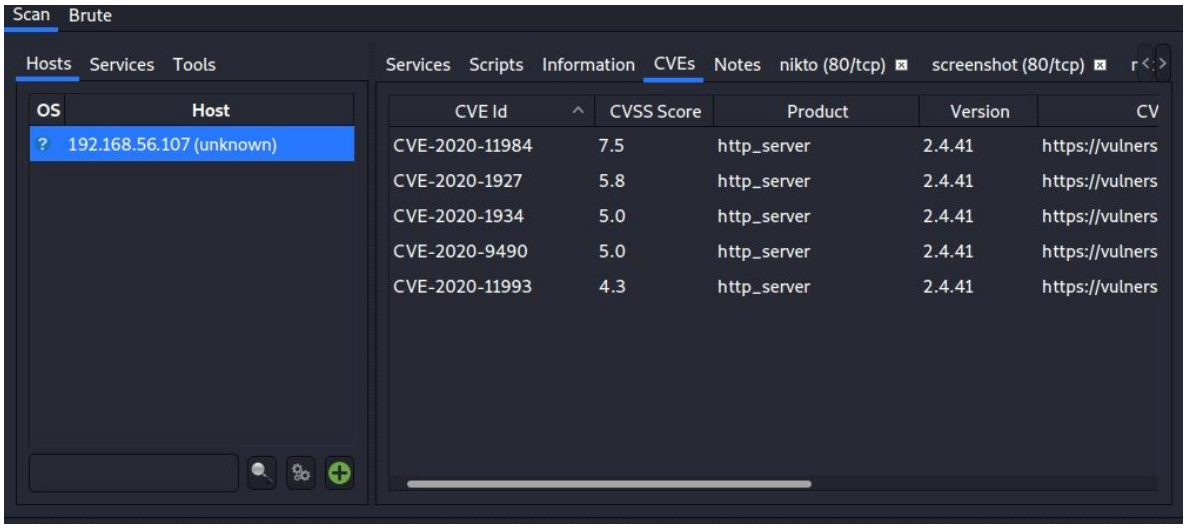
Ilustración 24 - *Escaneo de vulnerabilidad - Redis - Legion*



Fuente El autor

Legion sobre Cassandra

Ilustración 25 - *Escaneo de vulnerabilidad – Cassandra - Legion*



Fuente El autor

Legion proporciona un listado de las posibles vulnerabilidades que el prototipo pueda tener, estas amenazas hacen parte de la lista registrada por el estándar CVE que se encarga de recopilar las vulnerabilidades y amenazas de seguridad informática.

Los resultados de este escaneo muestran que las vulnerabilidades están presentes en los tres gestores de bases de datos y corresponden al puerto 80 es decir configuración http del servidor, estos resultados no implican que se pueda realizar un mismo ataque a las tres BDNR.

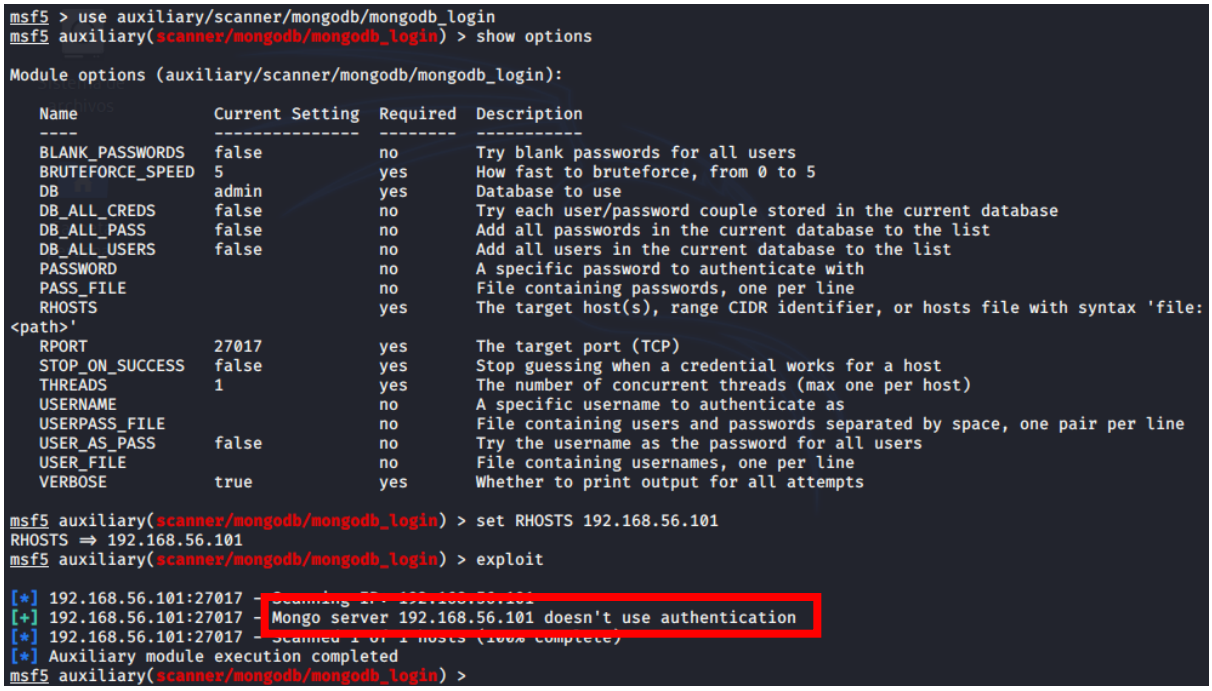
8.3.4 Explotación sobre prototipo

Una vez se tiene identificadas las vulnerabilidades se usa la herramienta Metasploit que es un framework que proporciona información acerca de la seguridad en un programa. Metasploit cuenta con una gran colección de exploits en las cuales tienen módulos que son los códigos que explotan las vulnerabilidades ya conocidas.

ATAQUES A MONGO (Metasploit)

- primer exploit: auxiliary/scanner/mongodb/mongodb_login

Ilustración 26 – Primer Exploit – MongoDB



Fuente El autor

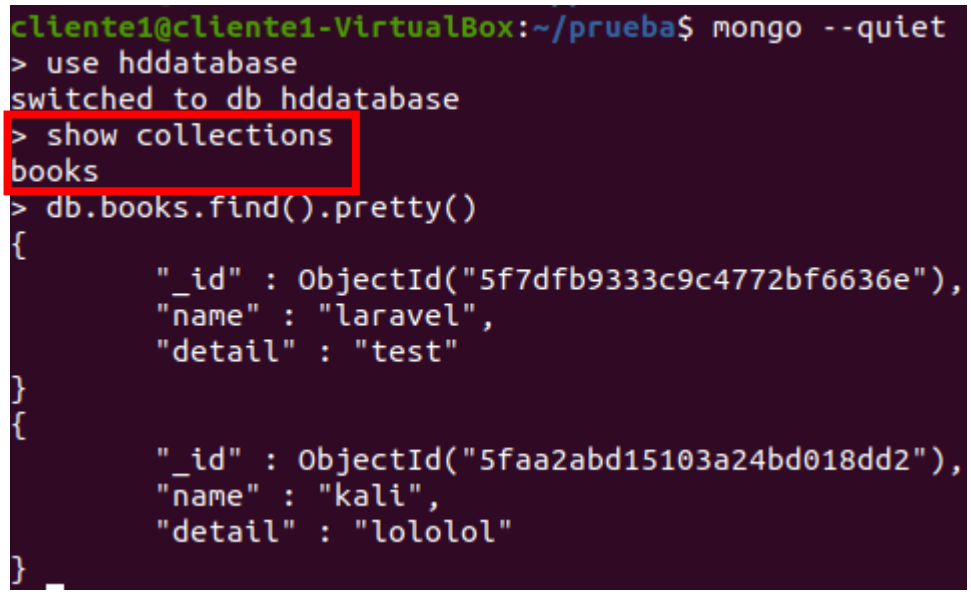
Se puede realizar un ataque considerando que el puerto de MongoDB está abierto y configurado por defecto en el 27017. Metasploit utiliza un script con un scanner para MongoDB en donde utiliza parámetros por defecto y en donde se ingresa la

dirección de la máquina donde se ejecuta el prototipo. Una vez ejecutado el ataque los resultados indican que el servidor de MongoDB no requiere autenticación lo que permite a cualquier usuario poder acceder a la base de datos del prototipo.

- Segundo exploit: exploit/linux/misc/mongod_native_helper

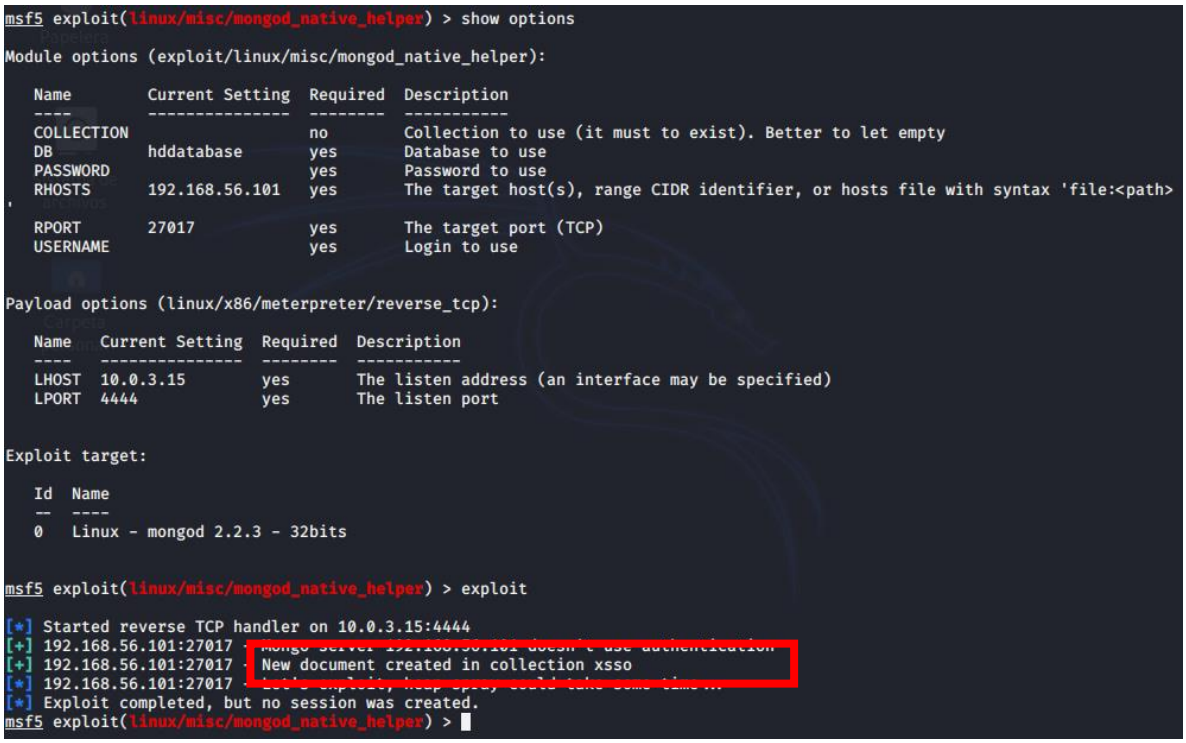
Durante este exploit se intenta agregar un registro a la base de datos principal llamada hddatabase en donde se almacena información del prototipo como se puede ver a continuación:

Ilustración 27 – Agregar Registro – MongoDB



Fuente El autor

Ilustración 28 – Segundo Exploit – MongoDB



Fuente El autor

El exploit se puede utilizar debido a que mongo está configurado por defecto, aunque si se conociera el usuario y la contraseña del administrador de la base de datos el exploit ofrece la opción de aplicarlo también. Para este caso solo es necesario asignarle el valor de la ip en el campo RHOST que no estaba previamente configurado. Como resultado Metasploit indica que ha generado un nuevo documento en la colección xsso (Nombre que es generado aleatoriamente en el script) en la base de datos hddatabase.

Ilustración 29 – Registro Verificado – MongoDB

Fuente El autor

Al verificar nuevamente en la base de datos comprobamos que se ha agregado una nueva colección y un nuevo elemento.

INYECCIÓN MANUAL EN MONGO

- **TAUTOLOGIAS**

Para realizar las pruebas de inyección se accede a la ventana del prototipo que realiza búsquedas en la base de datos y se ingresan elementos que modifiquen la consulta para realizar otra acción en la base de datos. Para este caso se inserta "[$ne]=1" sin comillas en la url de la página. De esta manera la consulta original que debería ser:

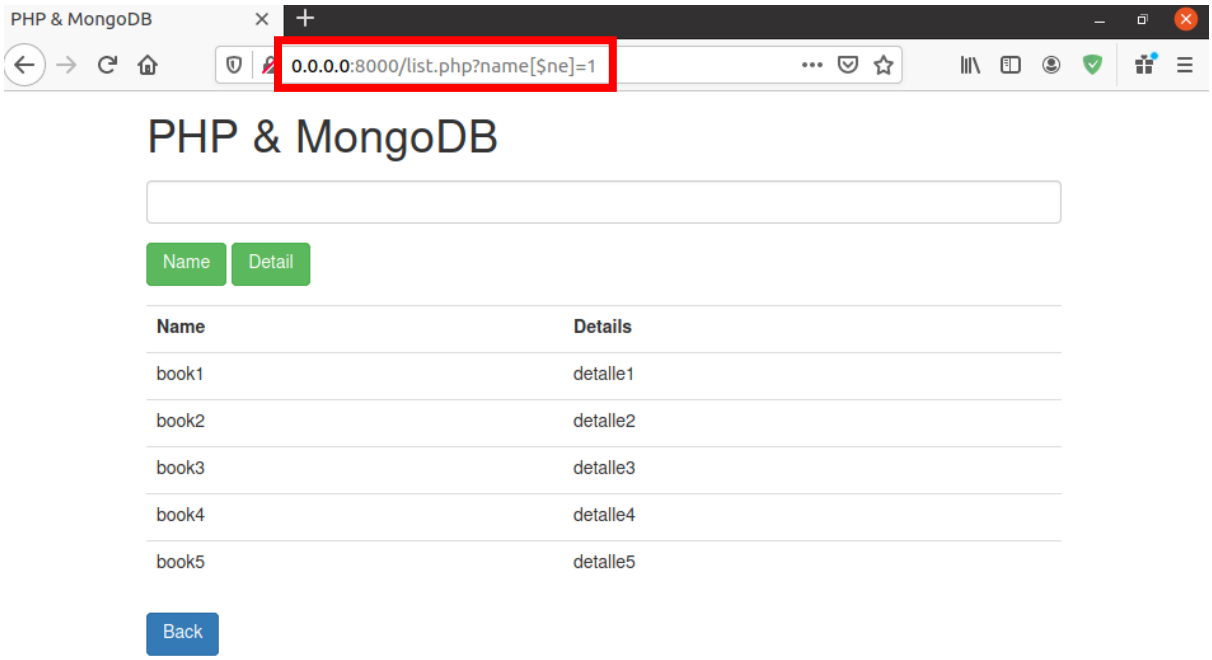
- $books = $collection->find(array('name'=>$_GET['name']))

Cambia por:

- $books = $collection->find(array('name'=>[$ne]=1))

Lo que se traduce en una búsqueda que lista todos los elementos en donde la variable name sea diferente de 1 que es la totalidad de elementos en el caso presentado.

Ilustración 30 – Inyección por Tautología – MongoDB



Fuente El autor

- **CONSULTAS INCORRECTAS**

El propósito de este tipo de inyección es generar errores en una consulta o un query para obtener un mensaje de error que comprometa información sobre la base de datos. Para este caso se inserta "[$$ne]=1" sin comillas en la url de la página. De esta manera la consulta original que debería ser:

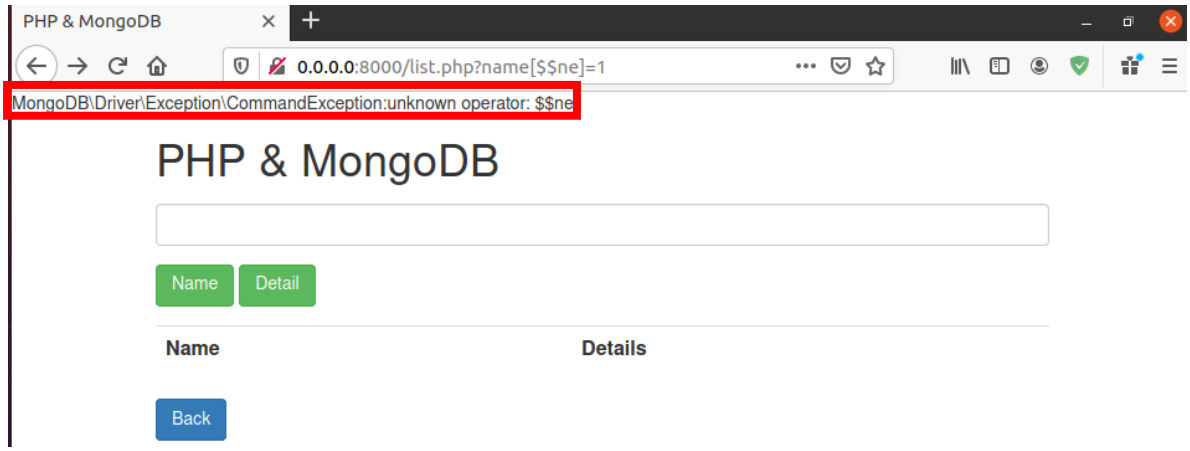
- $books = $collection->find(array('name'=>$_GET['name']))

Cambia por:

- $books = $collection->find(array('name'=>[$$ne]=1))

El mensaje de error muestra como resultado que la consulta se realizó en MongoDB y así se identifica es el gestor de bases de datos que utiliza el prototipo:

Ilustración 31 – Consultas Incorrectas – MongoDB



Fuente El autor

ATAQUES A REDIS (Metasploit)

- Primer exploit: auxiliary/scanner/redis/redis_login

Ilustración 32 – Primer Exploit – Redis

```
msf5 > use auxiliary/scanner/redis/redis_login
msf5 auxiliary(scanner/redis/redis_login) > show options

Module options (auxiliary/scanner/redis/redis_login):

   Name              Current Setting                                                Required  Description
   ----              ---------------                                                --------  -----------
   BLANK_PASSWORDS   false                                                          no        Try blank passwords for a
ll users
   BRUTEFORCE_SPEED  5                                                            yes       How fast to bruteforce, f
rom 0 to 5
   DB_ALL_PASS       false                                                        no        Add all passwords in the
current database to the list
   PASSWORD          foobared                                                     no        Redis password for authen
tication test
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt  no        The file that contains a
list of of probable passwords.
   RHOSTS                                                                         yes       The target host(s), range
 CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT             6379                                                         yes       The target port (TCP)
   STOP_ON_SUCCESS   false                                                        yes       Stop guessing when a cred
ential works for a host
   THREADS           1                                                            yes       The number of concurrent
threads (max one per host)
   VERBOSE           true                                                         yes       Whether to print output f
or all attempts

msf5 auxiliary(scanner/redis/redis_login) > set RHOSTS 192.168.56.101
RHOSTS => 192.168.56.101
msf5 auxiliary(scanner/redis/redis_login) > exploit

[+] 192.168.56.101:6379   - 192.168.56.101:6379   - Login Successful: redis:foobared (Successful: +OK
)
[*] 192.168.56.101:6379   - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

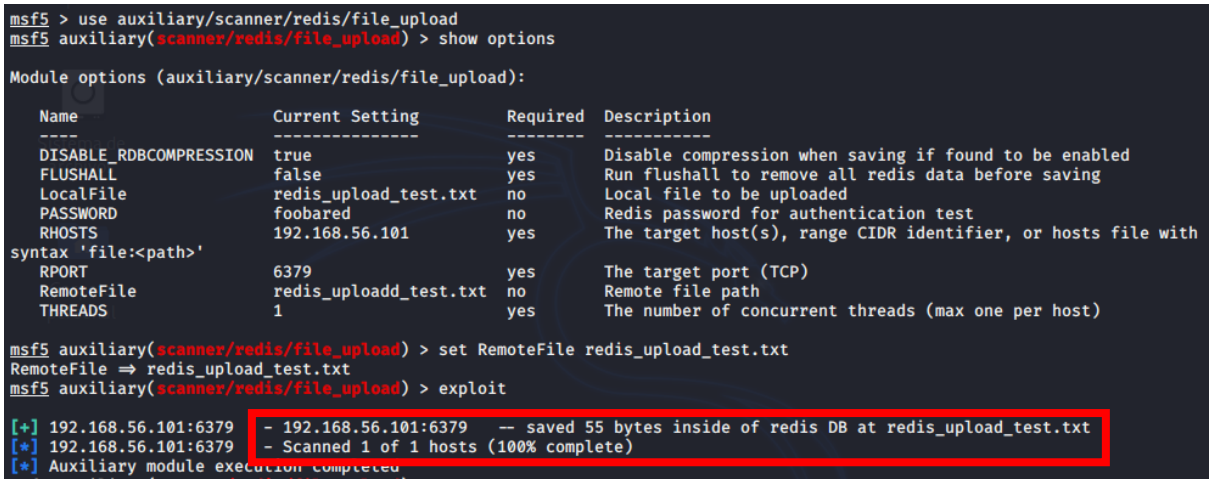
Fuente El autor

Se puede realizar un ataque considerando que el puerto de Redis está abierto y configurado por defecto en el 6379. Metasploit utiliza un script con un scanner para Redis en donde utiliza parámetros por defecto y en donde se ingresa la dirección de

la máquina donde se ejecuta el prototipo. Una vez ejecutado el ataque los resultados indican que el servidor de Redis requiere autenticación, pero al estar configurado por defecto el script descubre que la contraseña es “foobared” lo que permite a cualquier usuario poder acceder a la base de datos del prototipo.

- Segundo exploit: auxiliary/scanner/redis/file_upload

Ilustración 33 – Segundo Exploit – Redis

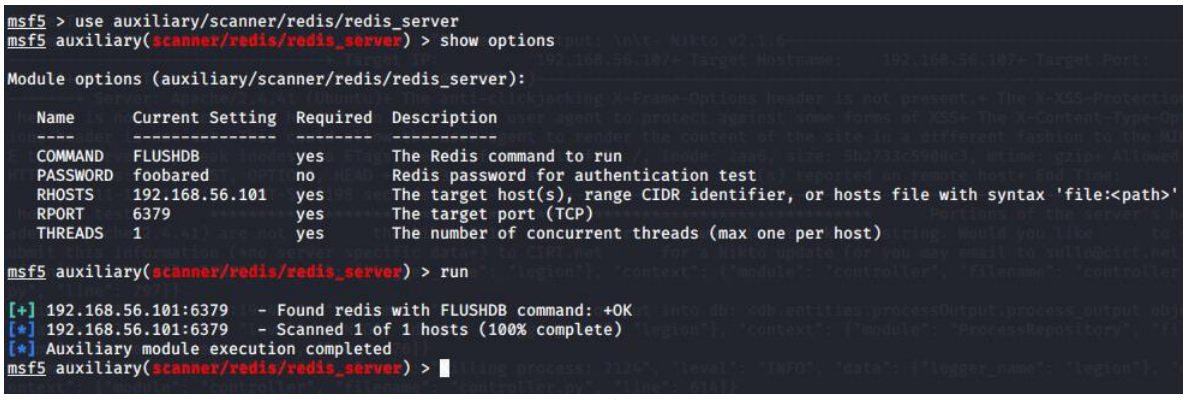


Fuente El autor

El siguiente exploit consiste en generar un archivo local que se replique en la base de datos de Redis debido a que este gestor tiene la capacidad de almacenar un gran tipo de datos incluidos archivos del formato txt esta vulnerabilidad puede implementar un archivo malicioso que capture y replique información comprometiendo la confidencialidad de los datos alojados en la base de datos.

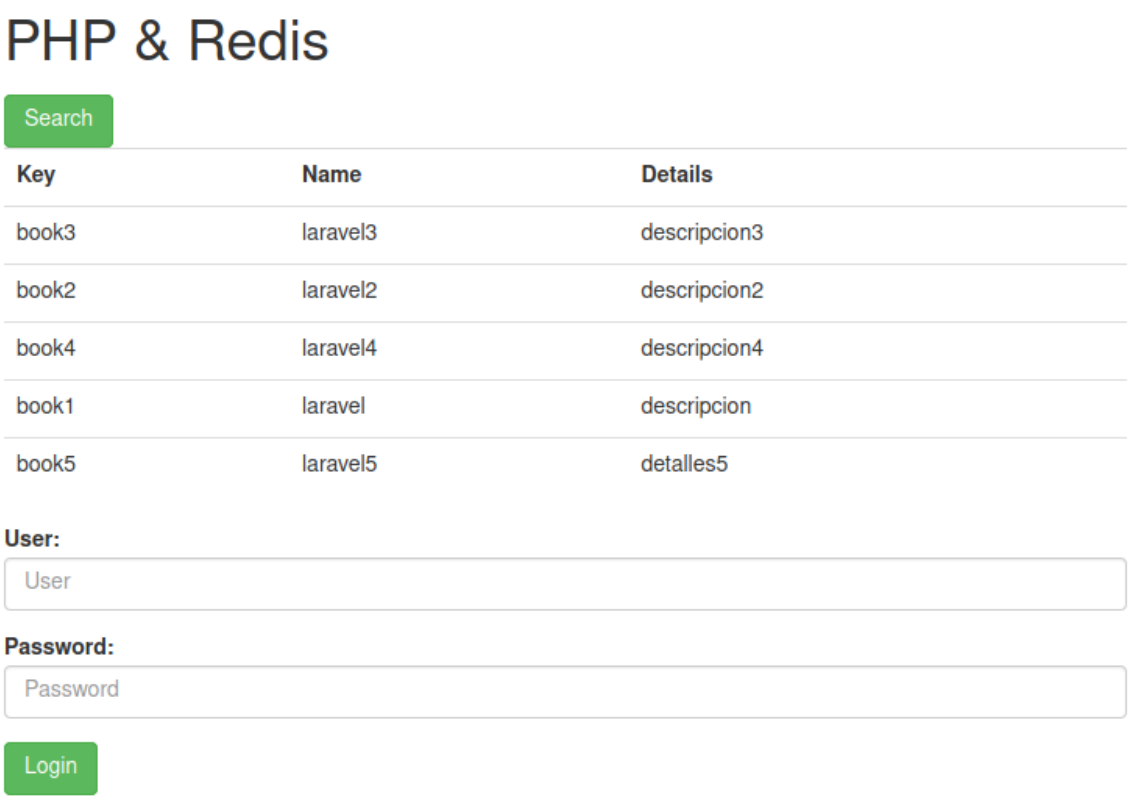
- Tercer exploit: auxiliary/scanner/redis/redis_server

Ilustración 34 – Tercer Exploit – Redis



Fuente El autor

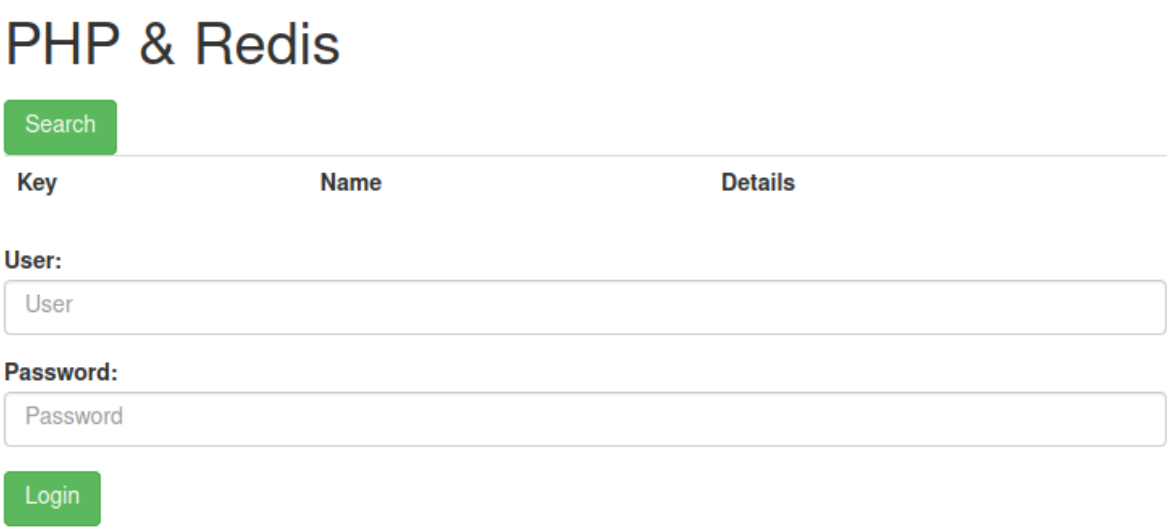
Ilustración 35 – Datos Registrados



Fuente El autor

Este exploit recibe como parámetro un valor que se asigna a la variable COMMAND en donde se ejecuta remotamente sobre la máquina víctima. Cuando se agrega el valor FLUSHDB a la variable el exploit ejecuta el comando en la consola de Redis y esta borra todos los registros existentes en la base de datos actual.

Ilustración 36 – Login



Fuente El autor

INYECCION EN REDIS

El protocolo Redis no tiene ningún concepto de escape de cadenas, por lo que la inyección es imposible en circunstancias normales utilizando una biblioteca cliente normal. El protocolo utiliza cadenas de longitud prefijada y es completamente binario seguro. (Redis Security, 2020)

ATAQUES A CASSANDRA (Metasploit)

- **EXPLOITS EN CASSANDRA**

La herramienta Metasploit fundamenta en parte sus scripts en los CVE que existen sobre los diferentes programas que donde se encuentran fallas de seguridad. Apache Cassandra cuenta con un manejo de consultas o queries basados en SQL y una flexibilidad notable a la hora de trabajar con java esta última ventaja se ve reflejada en los más recientes CVE que se han encontrado con respecto a las vulnerabilidades de seguridad debido a que ambos ataques consisten en ejecutar código java arbitrario de manera remota es así que para esta investigación no se encuentran exploits disponibles en la herramienta Metasploit ya que el prototipo desarrollado se compone de código PHP y la investigación se enfoca en los gestores y no en los lenguajes de programación.

INYECCIÓN EN CASANDRA

- **TAUTOLOGIAS**

Para realizar las pruebas de inyección se accede a la ventana del prototipo que realiza búsquedas en la base de datos y se ingresan elementos que modifiquen la consulta para realizar otra acción en la base de datos. Para este caso no es posible debido a dos situaciones.

El sistema de comunicación entre capas de Cassandra con PHP utiliza un framework de RPC (Remote Procedure Call) el cual no está basado en análisis de Strings así que no se pueden ejecutar dos consultas en una sola entrada. Por otra parte, al intentar ingresar más de un valor para una consulta debe asignarse primero a un arreglo y después enviar el arreglo a la consulta de esta manera los valores no se ejecutan directamente sobre la consulta como se ve en el siguiente caso:

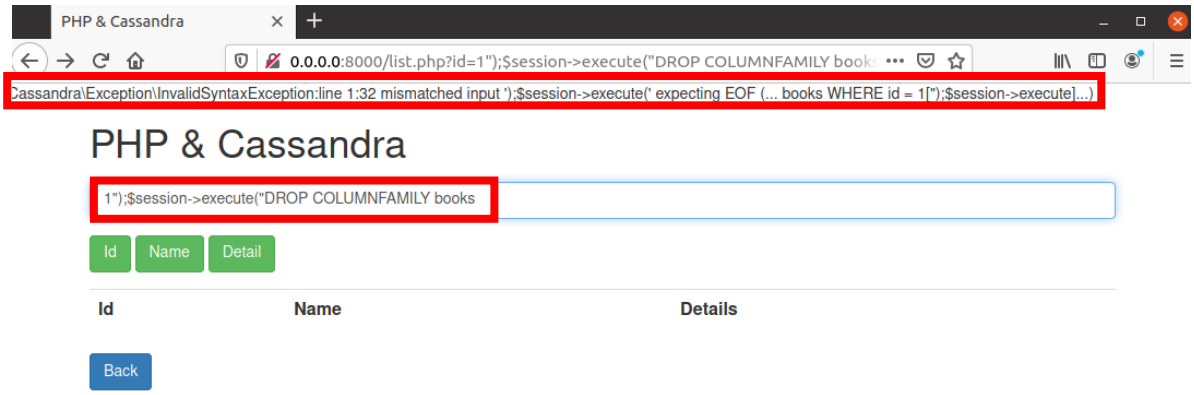
```
$options=array('arguments'=>array($_POST['code'],
                                  $_POST['name'],
                                  $_POST['detail'],));
$session->execute("INSERT INTO books (id, name, detail) VALUES (?,?,?)",$options);
```

Un intento por aplicar la inyección fue el siguiente y su resultado fue un error que no genero cambios como se esperaba:

$result = $session->execute ("SELECT * FROM books WHERE id = $_GET[id]");

1"); $session->execute ("DROP COLUMNFAMILY books

Ilustración 37 – Inyección por Tautología - Cassandra



Fuente El autor

- **CONSULTAS INCORRECTAS**

El propósito de este tipo de inyección es generar errores en una consulta o un query para obtener un mensaje de error que comprometa información sobre la base de datos. Para este caso se inserta 1"); para terminar una sentencia y agregar un error de la siguiente manera:

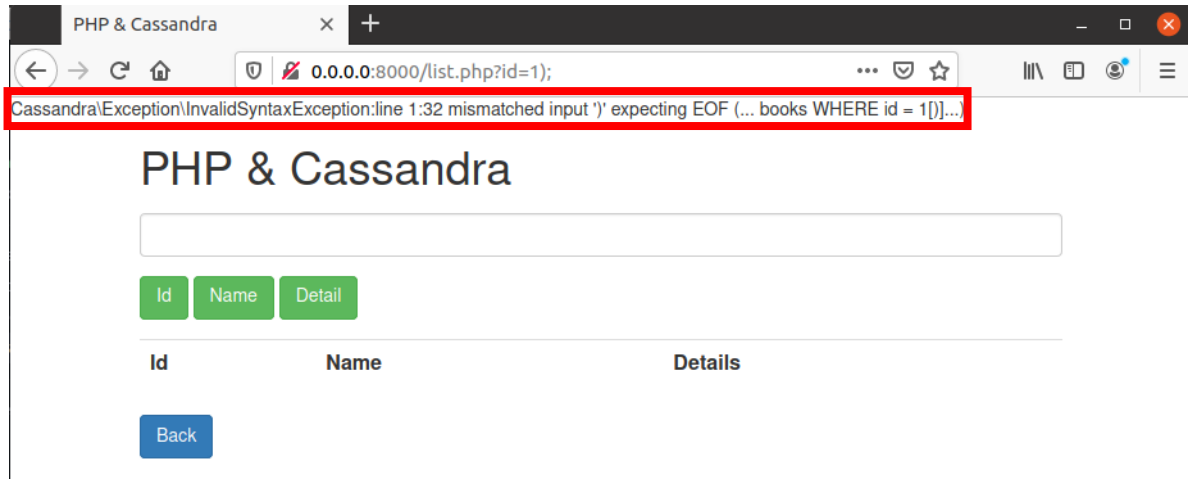
$result = $session->execute ("SELECT * FROM books WHERE id = $_GET['id']");

Cambia por:

$result = $session->execute ("SELECT * FROM books WHERE id = 1");");

El mensaje de error muestra como resultado que la consulta se realizó en Cassandra y así se identifica es el gestor de bases de datos que utiliza el prototipo:

Ilustración 38 – Consultas Incorrectas - Cassandra



Fuente El autor

8.4 Fase 4. Producción de informe.

8.4.1 Resultados

Los resultados de las pruebas se presentan en una tabla comparativa entre los gestores analizados y se compone de cada una de las herramientas que se utilizaron además de las pruebas de inyección NoSQL, que se implementaron manualmente. En la sección de análisis de resultados se detalla más a fondo los registros presentados en la tabla.

Tabla 21 - Análisis Resultados

	Mongo	**Redis**	**Cassandra**
Escaneo básico de Nmap	Escaneo de puertos 80, 8000 y 27017	Escaneo de puertos 80, 8000 y 6379	Escaneo de puertos 80, 8000, 7000, 9042 y 9160
Scripts de Nmap	Error en la aplicación de scripts	Script de información Script de fuerza bruta	Script de información Script de fuerza bruta
Errores más relevantes con Nikto	8000 y 27017 anti-clickjacking x-Frame-Options header no está presente. X-XSS-Protection header no está definido X-Content-Type-Options no está configurado config.php	8000 anti-clickjacking X-Frame-Options header no está presente. X-XSS-Protection header no está definido X-Content-Type-Options no está configurado config.php	8000 anti-clickjacking x-Frame-Options header no está presente. X-XSS-Protection header no está definido X-Content-Type-Options no está configurado config.php
Vulnerabilidades en Nessus y factores de riesgo	4 medio 38 información	1 alto 3 medio 38 información	4 medio 38 información
Resultados más relevantes con Legion	cve-2020-11984 cve-2020-1927 cve-2020-1934 cve-2020-9490 cve-2020-11903	cve-2020-11984 cve-2020-1927 cve-2020-1934 cve-2020-9490 cve-2020-11903	cve-2020-11984 cve-2020-1927 cve-2020-1934 cve-2020-9490 cve-2020-11903
Resultados de exploits utilizados en Metasploit	No usa autenticación Crea un nuevo elemento en la base de datos	Usa autenticación por defecto password=”foobared” Inserción de elementos en la base de datos	No disponible

		Control de la consola de comandos de Redis	
Inyección por tautología	Vulnerable	Protegido	Protegido
Inyección por consulta ilegal	Vulnerable	Vulnerable	Vulnerable

Fuente El autor

8.4.2 Análisis de resultados

Inicialmente por medio de la herramienta Nmap se realizó un solo escaneo a las redes que estén vinculadas junto a todos los puertos que estén abiertos, de esta manera se encuentran todos los puertos que utilizan todos los gestores. Cada gestor se halla configurados por defecto.

Existen scripts en Nmap que facilitan el trabajo en la fase de reconocimiento, para los tres gestores se encontraron scripts que permiten encontrar usuarios que hayan sido configurados por defecto para realizar ataques de fuerza bruta y también scripts que exponen la información básica de las bases de datos y aunque se encontró un script adicional, pero este se encuentra únicamente para MongoDB, esta base de datos fue la única sobre la cual los scripts no generaron ningún resultado.

Es de esperar que el análisis que ejecuta Nikto genere los mismos resultados para cada uno de los escaneos realizados, debido a que dicho análisis se ejecuta en mayor medida sobre un entorno web, también es lógico esperar este resultado ya que el prototipo maneja el mismo diseño para las tres bases de datos. También destaca que Nikto puede obtener resultados de un escaneo sobre el puerto por defecto de MongoDB, aunque los errores que expone son los mismos a los hallados en el prototipo, esto ocurre porque es el único de los puertos que maneja una interfaz directamente en el puerto 27017.

La herramienta Nessus resulto ser de particular utilidad aun en su versión gratuita debido a que cuenta con una gran cantidad de parámetros para la configuración de la búsqueda de vulnerabilidades entre las que se encuentran:

- Con la configuración de reconocimiento se puede seleccionar si se desea realizar escaneos sobre todos o una cantidad específica de puertos.

- Con la configuración de evaluación de seguridad, se puede seleccionar un análisis más completo en más tiempo o más general en un menor tiempo, esta configuración también aporta pruebas de ataques de fuerza bruta.
- La configuración avanzada está relacionada con la cantidad de intentos de acceso a una dirección ip o tiempo de cada fase de ataque.

En cada una de las fases descritas se encuentran opciones de configuración e incluso diferentes tipos de escaneo como; Avanzado, básico o de reconocimiento. Adicionalmente la capacidad de documentar automáticamente generando informes de los resultados representa una gran ventaja en términos de tiempo y esfuerzo.

Legion es una herramienta que ejecuta varios elementos de escaneo de seguridad de forma automática, Legion realiza los escaneos por medio de una selección predeterminada de scripts y comandos de Nmap y Nikto. Para este caso se decidió verificar los resultados obtenidos manualmente junto a los automatizados para que posteriormente sean comparados y así descartar falsos positivos.

El uso de Exploits, mediante la herramienta Metasploit, permite encontrar resultados importantes como lo son el detectar y generar accesos a las bases de datos por medio de fallos en la configuración de autenticación y la capacidad de controlar remotamente las consolas de comandos de los gestores analizados con base en los errores conocidos como CVE. En este caso se encontró que se puede acceder a la máquina de mongo y Redis,, en mongo es posible agregar nuevas colecciones y registros, en Redis se puede tener acceso directo a la consola de comandos lo que permite obtener información o borrar toda la información de la base de datos y finalmente lo más destacable es que en Cassandra no existe una sola vulnerabilidad a estos ataques en Metasploit y aun por medio de búsquedas externas no fue posible acceder para alterar o ver información privada de la base de datos.

Los últimos resultados obtenidos son sobre inyección NoSQL y se clasificaron en dos tipos de ataques:

- Tautología: El objetivo de este ataque es exponer información de la base de datos por medio de una sentencia que siempre es verdadera. Para este tipo de ataques solo se obtuvieron los resultados esperados sobre MongoDB debido a que utiliza el operador reservado de consulta $ne, sin embargo, en Redis y Cassandra no hay operadores que funcionen de manera similar y sus configuraciones no permiten generar más de dos consultas con un mensaje de entrada.

- Consulta ilegal: Este tipo de ataque es fácil de encontrar debido a que cuando se trabaja con un componente que permita la interacción entre un gestor de bases de datos y un lenguaje de programación, los errores muestran las direcciones completas desde donde ocurre el problema y en los casos presentados todos muestran el gestor de bases de datos que utiliza el prototipo.

8.4.3 Contramedidas

A continuación se presenta una guía con las contramedidas de las posibles vulnerabilidades. Inicia con métodos que permiten evitar ataques de inyección y este a su vez es aplicable a cualquiera de los tres prototipos conectados al respectivo gestor y finaliza con un instructivo de configuración para autenticación en los gestores de bases de datos.

Mitigación de errores de inyección:

Este tipo de ataques se pueden mitigar utilizando funciones del propio lenguaje de programación que parametricen o aseguren los datos que se están almacenando como la función filter_var() que es la recomendación de la página oficial de PHP o simplemente agregando una función a los elementos en donde reciban variable como se describe a continuación:

- $books = $collection->find(array('name'=> implode($_GET['name'])));
- $result = $session->execute ("SELECT * FROM books WHERE id = (int)$_GET['id']");

PHP implode(): Esta función permite convertir un array en una cadena de texto de una manera rápida y eficaz, así si se trata de almacenar más de un elemento no tendrá efecto.

El parámetro (int) establece que el valor almacenando en el $_GET['id'] sea un numero entero y no almacene un valor como texto.

Mitigación de errores de configuración

Para evitar accesos indeseados a la base de datos por medio de usuario y contraseñas configurados por defecto se debe agregar un nuevo usuario y contraseña y eliminar o remover los permisos de los usuarios por defecto como se describe a continuación:

MongoDB:

Para realizar la configuración correcta o segura en MongoDB se debe acceder a la línea de comandos e ingresar en orden.

```
$ mongo
> use admin
> db.createUser({
user: “newUser”,
pwd: “newPass”,
roles: [{ role: “userAdminAnyDatabase”, db: “admin” }] });
```

Así se crea un nuevo usuario administrador. El siguiente paso es reiniciar el servicio y acceder nuevamente a mongo, pero ahora con el usuario nuevo.

```
$ MongoDB-u “newUser” -p “newPass” – authenticationDatabase “admin”
```

Finalmente se accede al archivo MongoDB.conf y se activa la casilla que corresponda a la seguridad. En el caso del prototipo simplemente se debe eliminar el # de la línea que contiene: #auth = true

También se recomienda asignar una ip específica y otro puerto al existente por defecto.

Ilustración 39 – configuración - MongoDB

Fuente El autor

Una vez seguidos todos los pasos se pueden agregar usuarios a cada base de datos y no se expondrá información si no se accede con el usuario que tiene permisos.

Redis:

Para realizar la configuración correcta o segura en Redis se accede al archivo redis.conf y se activa la casilla que corresponda a la seguridad. En el caso del prototipo simplemente se debe eliminar el # de la línea que contiene:

#requirepass foobared

Luego se cambia la palabra foobared por la nueva contraseña. Existe una sugerencia en el archivo redis.conf que propone utilizar contraseñas seguras debido a que es vulnerable a ataques de fuerza bruta.

Ilustración 40 – Nueva Contraseña - Redis

```
# Warning: since Redis is pretty fast an outside user can try up to
# 150k passwords per second against a good box. This means that you should
# use a very strong password otherwise it will be very easy to break.
#
requirepass nuevacontraseña
```

Fuente El autor

También se recomienda asignar una ip específica y otro puerto al existente por defecto.

Cassandra:

Para realizar la configuración correcta o segura en Cassandra se debe acceder al archivo cassandra.yaml y modificar el tipo de autenticación de AllowAllAuthenticator a PasswordAuthenticator.

Ilustración 41 – Autenticación - Cassandra

```
# - AllowAllAuthenticator performs no checks - set it to disable authentication.
# - PasswordAuthenticator relies on username/password pairs to authenticate
#   users. It keeps usernames and hashed passwords in system_auth.roles table.
#   Please increase system_auth keyspace replication factor if you use this authenticator.
#   If using PasswordAuthenticator, CassandraRoleManager must also be used (see below)
authenticator: PasswordAuthenticator
```

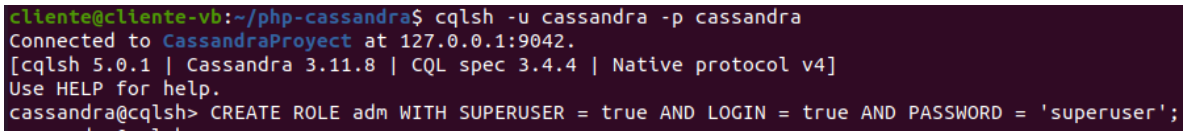
Fuente El autor

Una vez realizado el cambio y reiniciado el servicio, para poder acceder a la consola de comandos de Cassandra se necesita un usuario y una contraseña que están configurados por defecto y es “cassandra” en los campos usuario y contraseña.

Cuando se accede a la consola se crea un nuevo usuario con todos los permisos después se accede con el nuevo usuario y finalmente se modifican los permisos del usuario por defecto para que no tenga acceso.

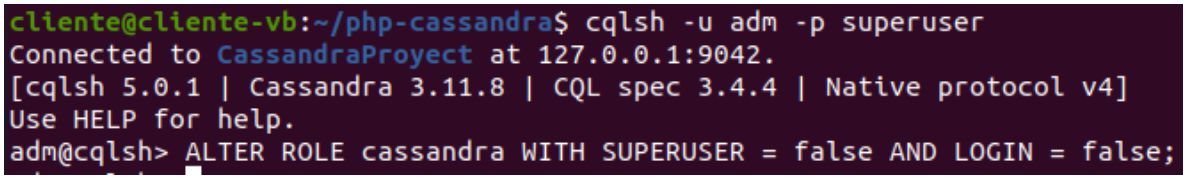
También se recomienda asignar una ip específica y otro puerto al existente por defecto.

Ilustración 42 – Abrir Terminal por Defecto



Fuente El autor

Ilustración 43 – Abrir Terminal con Nuevo Usuario



Fuente El autor

CONCLUSIONES

El fin del presente proyecto de grado consta en identificar vulnerabilidades de seguridad en BDNR, a partir de escenarios controlados y correctamente configurados para el uso de técnicas de pentesting ya que permiten comprobar la seguridad de las bases de datos a través de pruebas de intrusión para detectar posibles vulnerabilidades en la red y la aplicación.

MongoDB es el gestor de bases de datos más popular de los tres gestores utilizados en el proyecto, esto implica que hay una clara ventaja a la hora de buscar documentación o ejemplos. Debido a la gran cantidad de información disponible sobre este gestor es más fácil mitigar los problemas que se encontraron en los resultados anteriormente mencionados, sin embargo, se encuentran más fallas en su seguridad y esto se ve reflejado en los registros de CVE el cual indica que es el gestor que tiene examinadas fallas de seguridad más recientes y como se pudo evidenciar es posible agregar información en la base de datos.

Apache Cassandra tiene una significativa ventaja sobre Redis y MongoDB ya que no es vulnerable a la herramienta Metasploit, sus dos CVE más reciente son de los año 2015 y 2018 en donde ambos son errores de ejecución de código en Java y no se aplican en sus versiones más recientes la cual se utiliza en este proyecto, su lenguaje denominado CQL fundamentado en SQL es sencillo de aprender si se tienen nociones básicas de este último y su configuración no es vulnerable a ataques de inyección por lo cual no fue posible ejecutar un ataque en el presente proyecto. Por otra parte, aspectos como instalación, configuración y capacidad de trabajar con otros lenguajes de programación llegan a ser significativamente más complejos que el mismo proceso en los otros dos gestores, cabe destacar que encontrar información y documentación de Cassandra puede ser un proceso complicado debido a que es menor la cantidad de informes o investigaciones.

La instalación y configuración de Redis es la más sencilla que se realizó durante la investigación si bien Redis tiene CVE más antiguos que MongoDB y Cassandra, estas vulnerabilidades son más graves debido a que dan acceso directo a la consola de comandos, permite replicar información y obtener acceso más fácil de autenticación. También cabe destacar que existen siete Exploits orientados a este gestor en Metasploit donde cuatro son para Windows y los otros tres para Linux lo que lo vuelve más vulnerable a ataques. El concepto de llave valor que utiliza no es complicado de entender, sin embargo, aunque se pueda encontrar información

sobre casos generales de Redis la documentación en lenguajes como PHP se ve más limitada que en los otros gestores.

La investigación inicial permite conocer la técnica de ataque más apropiada para evaluar la seguridad en las bases de datos NoSQL, se le conoce como inyección NoSQL sin embargo a lo largo de la investigación se demuestra que este tipo de ataques no son compatibles con todos los gestores NoSQL debido al manejo de strings y protocolos como el RPC.

Durante la fase de investigación se encontraron una gran cantidad herramientas de análisis de seguridad para realizar escaneos de vulnerabilidad más completos, estas son:

- Legión: Implementa scripts de Nmap.
- Nikto: Ofrece alternativas de scripts manuales adicionales
- Nessus: Parametriza en gran medida la configuración para reconocimiento de vulnerabilidades y genera documentos de resultados de auditoría de manera automatizada.

PHP en su versión 7.4.11 dispone de paquetes que permiten interactuar con HTML5, además de contar con una curva de aprendizaje baja, esto quiere decir que la sintaxis sea sencilla de utilizar. Se concluyó que PHP es perfectamente compatible en 2 de los gestores de bases de datos en sus versiones más recientes que son MongoDB 3.6.8 y Redis 5.0.7, por otro lado, PHP no es compatible con la versión más reciente de Cassandra 3.11.8, se debe usar una versión más antigua para trabajar con Cassandra.

MongoDB al ser una BDNR orientada a documentos, es bastante vulnerable a ataques por Inyección NoSQL como Tautologías y consultas ilegales. Al revisar y comparar los resultados se concluyó que MongoDB es el gestor de bases de datos NoSQL más vulnerable y más propenso a ataques de pentesting.

RECOMENDACIONES

Dados los resultados y la metodología empleada en esta investigación podrían realizarse nuevos estudios con las mismas pruebas para comparar la seguridad de gestores de bases de datos en diferentes lenguajes de programación y actualizar los resultados dependiendo de las versiones que utilicen MongoDB, Redis o Cassandra para trabajar con PHP.

También se pueden actualizar los resultados dependiendo de si existen nuevas herramientas, por ejemplo, la herramienta NoSQLMap no pudo ser utilizada en esta investigación debido a que aún no se han implementado funciones para interactuar con Redis y Cassandra directamente.

Con los resultados de las búsquedas de herramientas para analizar, realizar ataques o reconocimiento se encontró que es viable desarrollar una nueva herramienta que permita interactuar específicamente a gestores de bases de datos NoSQL.

BIBLIOGRAFÍA

Adastra, V., 2011. *Conceptos Básicos De Nikto – Técnicas De Escaneo De Servidores Y Aplicaciones Web*. [online] Seguridad en Sistemas y Técnicas de Hacking. TheHackerWay (THW). Available at: <https://thehackerway.com/2011/05/12/conceptos-basicos-de-nikto-tecnicas-de-escaneo-de-servidores-y-aplicaciones-web/> [Accessed 13 November 2020].

Amazon, 2020. Bases De Datos No Relacionales | Bases De Datos De Gráficos | AWS. [online] Amazon Web Services, Inc. Obtenido de: <https://aws.amazon.com/es/nosql/> [Accesible 4 June 2020].

Amazon, 2020. *¿Qué Es Redis? – Amazon Web Services (AWS)*. [online] Amazon Web Services, Inc. *Obtenido de:* <https://aws.amazon.com/es/elasticache/what-is-redis/> *[Accesible 4 June 2020].*

AWS, 2020. *¿Qué Es AWS?*. [online] Amazon Web Services, Inc. Available at: <https://aws.amazon.com/es/what-is-aws/> [Accessed 4 August 2020].

Barrera, A., 2020. *JSON: ¿Qué Es Y Para Qué Sirve?*. [online] NextU LATAM. Available at: <https://www.nextu.com/blog/que-es-json/> [Accessed 4 August 2020].

BD-Engines, 2020. *Historical Trend Of The Popularity Ranking Of Database Management Systems*. [online] Db-engines.com. *Obtenido de:* <https://db-engines.com/en/ranking_trend> [*Accesible 6 June 2020*].

Blogger, 2013. *Metodologías Y Herramientas De Ethical Hacking*. [online] Seguridadinformaticahoy.blogspot.com. *Obtenido de:* <https://seguridadinformaticahoy.blogspot.com/2013/02/metodologias-y-herramientas-de-ethical.html> *[Accesible 4 June 2020].*

Cassandra, A., 2016. Apache Cassandra. [online] Cassandra.apache.org. Obtenido de: <https://cassandra.apache.org/> [Accesible 4 June 2020].

CVE, D., *2390*. *CVE-2019-2390 : An Unprivileged User Or Program On Microsoft Windows Which Can Create Openssl Configuration Files In A Fixed Location Ma.* [online] Cvedetails.com. *Obtenido de:* <https://www.cvedetails.com/cve/CVE-2019-2390/> *[Accesible 4 June 2020].*

CVE, D., 2019. *CVE-2019-2389 : Incorrect Scoping Of Kill Operations In Mongodb Server's Packaged Sysv Init Scripts Allow Users With Write Access T*. [online] Cvedetails.com. Available at: <https://www.cvedetails.com/cve/CVE-2019-2389/> [Accessed 31 August 2020].

CVE, D., 2019. *CVE-2019-2386 : After User Deletion In Mongodb Server The Improper Invalidation Of Authorization Sessions Allows An Authenticated User&#*. [online] Cvedetails.com. Available at: <https://www.cvedetails.com/cve/CVE-2019-2386/> [Accessed 31 August 2020].

CVE, D., *8016*. *CVE-2018-8016 : The Default Configuration In Apache Cassandra 3.8 Through 3.11.1 Binds An Unauthenticated JMX/RMI Interface To All Netwo*. [online] Cvedetails.com. *Obtenido de:* <https://www.cvedetails.com/cve/CVE-2018-8016/> *[Accesible 4 June 2020].*

CVE, D., 2015. *CVE-2015-0225 : The Default Configuration In Apache Cassandra 1.2.0 Through 1.2.19, 2.0.0 Through 2.0.13, And 2.1.0 Through 2.1.3 Binds*. [online] Cvedetails.com. Available at: <https://www.cvedetails.com/cve/CVE-2015-0225/> [Accessed 31 August 2020].

CVE, D., *10193*. *CVE-2019-10193 : A Stack-Buffer Overflow Vulnerability Was Found In The Redis Hyperloglog Data Structure Versions 3.X Before 3.2.13, 4.X*. [online] Cvedetails.com. *Obtenido de:* <https://www.cvedetails.com/cve/CVE-2019-10193/> *[Accesible 4 June 2020].*

CVE, D., 2018. *CVE-2018-12453 : Type Confusion In The Xgroupcommand Function In T_Stream.C In Redis-Server In Redis Before 5.0 Allows Remote Attackers T*. [online] Cvedetails.com. Available at: <https://www.cvedetails.com/cve/CVE-2018-12453/> [Accessed 31 August 2020].

CVE, D., 2018. *CVE-2018-12326 : Buffer Overflow In Redis-Cli Of Redis Before 4.0.10 And 5.X Before 5.0 RC3 Allows An Attacker To Achieve Code Execution*. [online] Cvedetails.com. Available at: <https://www.cvedetails.com/cve/CVE-2018-12326/> [Accessed 31 August 2020].

Castro, C., 2015. *¿Qué Es El Ethical Hacking? - BLOG | UTEL*. [online] BLOG | UTEL. Available at: <https://www.utel.edu.mx/blog/menu-profesional/que-es-el-ethical-hacking/> [Accessed 4 August 2020].

Carles, J., 2013. *Firewall. Que Es, Para Que Sirve, Como Funciona, Tiene Limitaciones?*. [online] geekland. Available at: <https://geekland.eu/que-es-y-para-que-sirve-un-firewall/> [Accessed 4 August 2020].

Carisio, E., 2020. *Qué Es El Diseño De Base De Datos Y Cómo Planificarlo*. [online] #ADNCLOUD. Available at: <https://blog.mdcloud.es/que-es-el-diseno-de-base-de-datos-y-como-planificarlo/> [Accessed 4 August 2020].

Catoira, F., 2013. *Nueva Funcionalidad De Monitoreo Continuo De Nessus | Welivesecurity*. [online] WeLiveSecurity. Available at: <https://www.welivesecurity.com/la-es/2013/06/17/nueva-funcionalidad-de-monitoreo-continuo-de-nessus/#:~:text=sobre%20sus%20redes.-,Nessus%20es%20una%20herramienta%20desarrollada%20para%20realizar%20escaneos%20de%20seguridad,de%20seguridad%20de%20los%20sistemas.> [Accessed 13 November 2020].

Cso, 2018. *¿Qué Es Wireshark? Así Funciona La Nueva Tendencia Esencial En Seguridad*. [online] Cso.computerworld.es. Available at: <https://cso.computerworld.es/tendencias/que-es-wireshark-asi-funciona-la-nueva-tendencia-esencial-en-seguridad> [Accessed 4 August 2020].

DATA, C., 2015. *¿Qué Importancia Tienen Las Bases De Datos A Nivel Empresarial?*. [online] DataCentric.*Obtenido de:* <https://www.datacentric.es/blog/bases-datos/importancia-bases-de-datos-2/> *[Accesible 4 June 2020].*

Dictionary, C., 2020. *DATA | Meaning In The Cambridge English Dictionary*. [online] Dictionary.cambridge.org. *Obtenido de:* <https://dictionary.cambridge.org/dictionary/english/data#dataset-cald4> *[Accesible 4 June 2020].*

EASSA, Ahmed M., et al. NoSQL Injection Attack Detection in Web Applications Using RESTful Service. *Programming and Computer Software*, 2018, vol. 44, no 6, p. 435-444.*Obtenido de:* <https://link.springer.com/article/10.1134/S036176881901002X> *[Accesible 4 June 2020].*

Ecured.cu. 2020. *Atributo (Informática) - Ecured*. [online] Available at: <https://www.ecured.cu/Atributo_(inform%C3%A1tica)> [Accessed 4 August 2020].

Erb, M., 2020. *7. Análisis De Riesgo*. [online] Gestión de Riesgo en la Seguridad Informática. Available at: <https://protejete.wordpress.com/gdr_principal/analisis_riesgo/> [Accessed 4 August 2020].

Everac99, 2008. *Alta Disponibilidad: Qué Es Y Cómo Se Logra*. [online] ::everac99. Available at: <https://everac99.wordpress.com/2008/08/19/alta-disponibilidad-que-es-y-como-se-logra/> [Accessed 4 August 2020].

EYMAR SILVA, 2017. [online] Repository.unad.edu.co. Available at: <https://repository.unad.edu.co/bitstream/handle/10596/14277/74375013.pdf?sequence=1&isAllowed=y> [Accessed 27 August 2020].

First, 2020. *CVSS V3.1 Specification Document*. [online] FIRST — Forum of Incident Response and Security Teams. *Obtenido de:* <https://www.first.org/cvss/v3.1/specification-document> *[Accesible 4 June 2020].*

Guru99, 2020. *What Is Database? What Is SQL?*. [online] Guru99.com. *Obtenido de:* <https://www.guru99.com/introduction-to-database-sql.html> *[Accesible 4 June 2020].*

Guru99, 2020. *What Is Mongodb? Introduction, Architecture, Features & Example*. [online] Guru99.com. *Obtenido de:* <https://www.guru99.com/what-is-mongodb.html> *[Accesible 4 June 2020].*

GUPTA, Shrankhla; SINGH, Nikhil Kumar; TOMAR, Deepak Singh. Analysis of NoSQL Database Vulnerabilities. En *Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT)*. 2018. p. 26-27. *Obtenido de:* <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3172769> *[Accesible 4 June 2020].*

Hou, B., Qian, K., Li, L. and Shi, Y., 2016. *Mongodb Nosql Injection Analysis And Detection - IEEE Conference Publication*. [online] Ieeexplore.ieee.org. *Obtenido de:* <https://ieeexplore.ieee.org/abstract/document/7545900/authors#authors> *[Accesible 4 June 2020].*

IICS, 2020. *Concientización De Ciberseguridad Seguridad Informática De Información*. [online] IICS. Available at: <https://www.iicybersecurity.com/concientizacion-ciberseguridad-seguridad-informatica.html> [Accessed 4 August 2020].

INFOSEGUR, 2013. *Confidencialidad – Seguridad Informática*. [online] Seguridad Informática. Available at: <https://infosegur.wordpress.com/tag/confidencialidad/> [Accessed 4 August 2020].

Intellipaat, 2016. *What Is Cassandra - An Introduction To Apache Cassandra*. [online] Intellipaat Blog. *Obtenido de:* <https://intellipaat.com/blog/what-is-apache-cassandra/> *[Accesible 4 June 2020].*

Java T Point, 2011. *DBMS SQL Introduction - Javatpoint*. [online] www.javatpoint.com. *Obtenido de:* <https://www.javatpoint.com/dbms-sql-introduction> *[Accesible 4 June 2020].*

Jonathan P, 2015. [online] Apps.dtic.mil. Available at: <https://apps.dtic.mil/sti/pdfs/AD1009288.pdf> [Accessed 27 August 2020].

Juntadeandalucia, 2020. *Conceptos Sobre La Escalabilidad | Marco De Desarrollo De La Junta De Andalucía*. [online] Juntadeandalucia.es. Available at: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/220#:~:text=Se%20entiende%20por%20escalabilidad%20a,n%C3%BAmero%20de%20usuarios%20del%20mismo.> [Accessed 4 August 2020].

Malwarebytes, 2020. *¿Qué Es El Malware? | Malwarebytes*. [online] Malwarebytes. Available at: <https://es.malwarebytes.com/malware/> [Accessed 4 August 2020].

Marin de la Fuente, J., 2019. *¿Qué Es Nmap? Por Qué Necesitas Este Mapeador De Red*. [online] José Marin de la Fuente. Available at: <https://www.marindelafuente.com.ar/que-es-nmap-por-que-necesitas-este-mapeador-de-red/> [Accessed 13 November 2020].

Microsoft, 2019. *Tablas - SQL Server*. [online] Docs.microsoft.com. Available at: <https://docs.microsoft.com/es-es/sql/relational-databases/tables/tables?view=sql-server-ver15#:~:text=Las%20tablas%20son%20objetos%20de,que%20contienen%20todos%20sus%20datos.&text=En%20las%20tablas%2C%20los%20datos,un%20campo%20dentro%20del%20registro.> [Accessed 4 August 2020].

MinTic, 2012. [online] Mintic.gov.co. Available at: <https://www.mintic.gov.co/portal/604/articles-4274_documento.pdf> [Accessed 28 October 2020].

MongoDB, 2020. *Nosql Databases Explained*. [online] MongoDB. *Obtenido de:* <https://www.mongodb.com/nosql-explained> *[Accesible 4 June 2020].*

MongoDB Alerts. 2020. *Mongodb Alerts*. [online] MongoDB. *Obtenido de:* <https://www.mongodb.com/alerts> *[Accesible 4 June 2020].*

Neuvoo.com.mx. 2020. *¿Qué Hace Un Administrador De Bases De Datos?*. [online] Available at: <https://neuvoo.com.mx/neuvooPedia/es/administrador-de-bases-de-datos/> [Accessed 4 August 2020].

Oracle, 2020. *Privilegios (Descripción General) - Guía De Administración Del Sistema: Servicios De Seguridad*. [online] Docs.oracle.com. Available at: <https://docs.oracle.com/cd/E24842_01/html/E23286/prbac-2.html> [Accessed 4 August 2020].

Owasp, 2017. [online] Wiki.owasp.org. Available at: <https://wiki.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf> [Accessed 31 August 2020].

Quanti, 2020. *¿Qué Es Load Balance O Balance De Carga?*. [online] Quanti. Available at: <https://quanti.com.mx/2018/07/20/que-es-load-balance-o-balance-de-carga/> [Accessed 20 July 2018].

Perez Prieto, J., 2020. *Estilo De Codificación Y Buenas Prácticas — Documentación De Curso De Python Para Astronomia - 20191118*. [online] Research.iac.es. Available at: <http://research.iac.es/sieinvens/python-course/source/estilo.html> [Accessed 4 August 2019].

Pagnotta, S., 2017. *Continúa La Ola De Ataques Extorsivos Contra Bases De Datos Mongodb | Welivesecurity.* [online] WeLiveSecurity. *Obtenido de:* <https://www.welivesecurity.com/la-es/2017/09/07/ataques-extorsivos-bases-de-datos-mongodb/> *[Accesible 4 June 2020].*

Poggi, N., 2018. *24 Estadísticas De Seguridad Informática Que Importan En El 2019 - The Missing Report.* [online] The Missing Report. *Obtenido de:* <https://preyproject.com/blog/es/24-estadisticas-seguridad-informatica-2019/> *[Accesible 4 June 2020].*

Raffino, M., 2020. *Dato En Informática: Concepto, Tipos Y Ejemplos*. [online] Concepto.de. *Obtenido de:* <https://concepto.de/dato-en-informatica/> *[Accesible 4 June 2020].*

Redis, 2020. *Redis*. [online] Redis.io. *Obtenido de:* <https://redis.io/> *[Accesible 4 June 2020].*

Redis Security, 2020. *Redis Security – Redis.* [online] Redis.io. Available at: <https://redis.io/topics/security#string-escaping-and-nosql-injection> [Accessed 17 November 2020].

RavenDB, 2020. *Why Are So Many Nosql Databases Getting Hacked? | Ravendb Nosql*. [online] Ravendb.net. *Obtenido de:* <https://ravendb.net/articles/why-are-so-many-nosql-databases-getting-hacked> [Accessed 5 June 2020].

Rouse, M., 2020. *What Is A Database Management System? - Definition From Whatis.Com*. [online] SearchSQLServer. *Obtenido de:* <https://searchsqlserver.techtarget.com/definition/database-management-system> *[Accesible 4 June 2020].*

ScaleGrid, 2019. *2019 Database Trends - SQL Vs. Nosql, Top Databases, Single Vs. Multiple Database Use*. [online] Scalegrid.io. *Obtenido de:* <https://scalegrid.io/blog/2019-database-trends-sql-vs-nosql-top-databases-single-vs-multiple-database-use/> *[Accesible 4 June 2020].*

Stack Overflow, 2017. *Stack Overflow Developer Survey 2017*. [online] Stack Overflow. *Obtenido de:* <https://insights.stackoverflow.com/survey/2017/>*[Accesible 4 June 2020].*

Studytonight, 2020. *Components Of DBMS (Database Management System) | Studytonight*. [online] Studytonight.com. *Obtenido de:* <https://www.studytonight.com/dbms/components-of-dbms.php> *[Accesible 4 June 2020].*

SQLcourse, 2019. *2019 Database Trends - SQL Vs. Nosql, Top Databases, Single Vs. Multiple Database Use*. [online] Scalegrid.io. *Obtenido de:* <https://scalegrid.io/blog/2019-database-trends-sql-vs-nosql-top-databases-single-vs-multiple-database-use/>*[Accesible 4 June 2020].*

Segurinformacion, 2018. *Seguridad De Redes Y Contramedidas*. [online] Sites.google.com. Available at: <https://sites.google.com/site/segurinformacion/> [Accessed 4 August 2020].

TechNet, E., 2016. *¿Cómo Mejorar El Rendimiento De Mi Base De Datos?*. [online] Executrain.com.mx. Available at: <https://www.executrain.com.mx/blog/microsoft/item/como-mejorar-el-rendimiento-de-mi-base-de-datos> [Accessed 4 August 2020].

Techopedia, 2019. *What Is A Database Management System (DBMS)? - Definition From Techopedia*. [online] Techopedia.com. *Obtenido de:* <https://www.techopedia.com/definition/24361/database-management-systems-dbms> *[Accesible 4 June 2020].*

TUNGGAL, A., 2020. What Is A Vulnerability?. [online] Upguard.com. Obtenido de: <https://www.upguard.com/blog/vulnerability> [Accesible 4 June 2020].

TUNGGAL, A., 2020. *What Is A Cyber Threat?*. [online] Upguard.com. *Obtenido de:* <https://www.upguard.com/blog/cyber-threat> *[Accesible 4 June 2020].*

Underc0de, 2019. *Legion Una Herramienta De Prueba De Penetración De Red Fácil De Usar*. [online] Underc0de.org. Available at: <https://underc0de.org/foro/hacking/t38960/#:~:text=Legion%20Una%20herramienta%20de%20prueba%20de%20penetraci%C3%B3n%20de%20red%20f%C3%A1cil%20de%20usar,-en%3A%20Marzo%2007&text=Legion%2C%20una%20bifurcaci%C3%B3n%20de%20la,explotaci%C3%B3n%20de%20sistemas%20de%20informaci%C3%B3n.> [Accessed 13 November 2020].

Wiki, 2020. *Kali Linux*. [online] Es.wikipedia.org. Available at: <https://es.wikipedia.org/wiki/Kali_Linux> [Accessed 4 August 2020].

Wiki, 2020. *Escáner De Vulnerabilidades*. [online] Es.wikipedia.org. Available at: <https://es.wikipedia.org/wiki/Esc%C3%A1ner_de_vulnerabilidades> [Accessed 4 August 2020].

wordpress, p., 2020. *6. Amenazas Y Vulnerabilidades*. [online] Gestión de Riesgo en la Seguridad Informática. Available at: <https://protejete.wordpress.com/gdr_principal/amenazas_vulnerabilidades/> [Accessed 4 August 2020].

Zaforas, M., 2016. *Cassandra, La Dama De Las Bases De Datos Nosql*. [online] Paradigmadigital.com. Available at: <https://www.paradigmadigital.com/dev/cassandra-la-dama-de-las-bases-de-datos-nosql/> [Accessed 4 August 2020].

ANEXOS

Anexo # 1 – Instalación y configuración MV

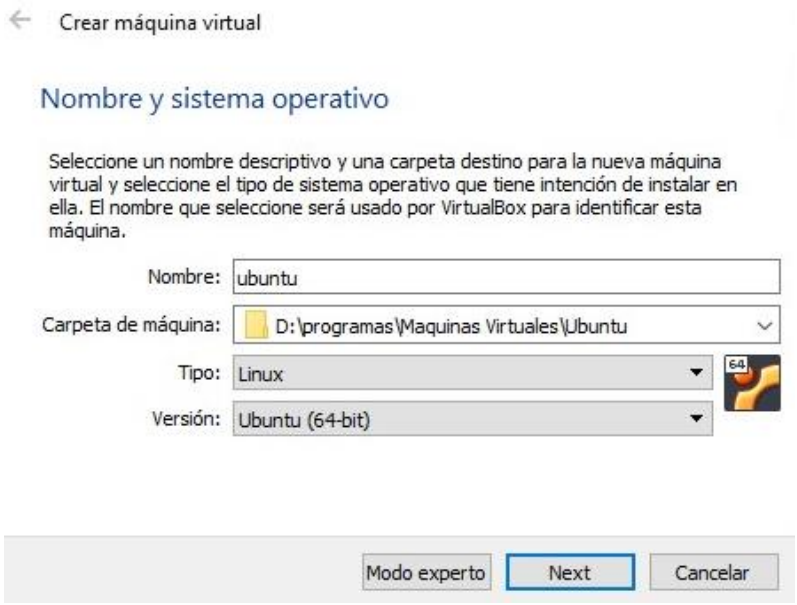
1) Máquina Virtual Ubuntu 20.04 y Kali Linux

Se debe contar con una máquina virtual que funcione como la víctima a la cual se le realizaran los ataques, se puede efectuar el mismo proceso para el servidor Kali Linux que tendría la función de atacante.

Paso 1: Instalación de la máquina virtual

Detalladamente se explicará el proceso de instalación de un servidor de Ubuntu en su versión 20.04, Primero debemos seleccionar un nombre descriptivo y una carpeta de destino para la nueva máquina virtual

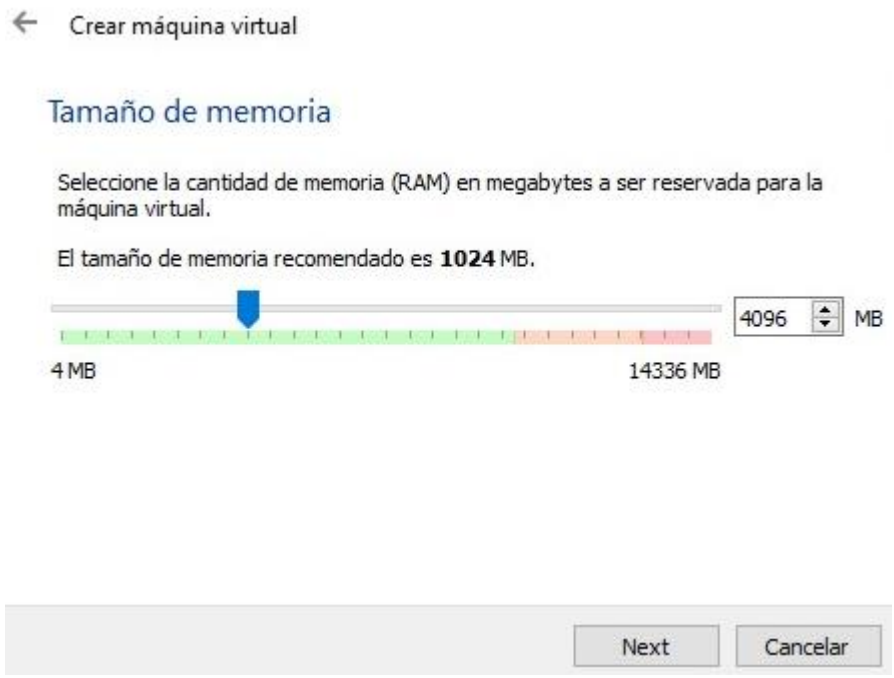
Ilustración 44 – Nombre y Sistema operativo



Fuente El autor

Debemos seleccionar la capacidad de memoria (RAM)

Ilustración 45 – Capacidad de RAM



Fuente El autor

Si desea puede añadir un disco duro virtual a la nueva máquina, para este proyecto se seleccionó la opción de crear un disco duro virtual

Ilustración 46 – Añadir un Disco Duro

Fuente El autor

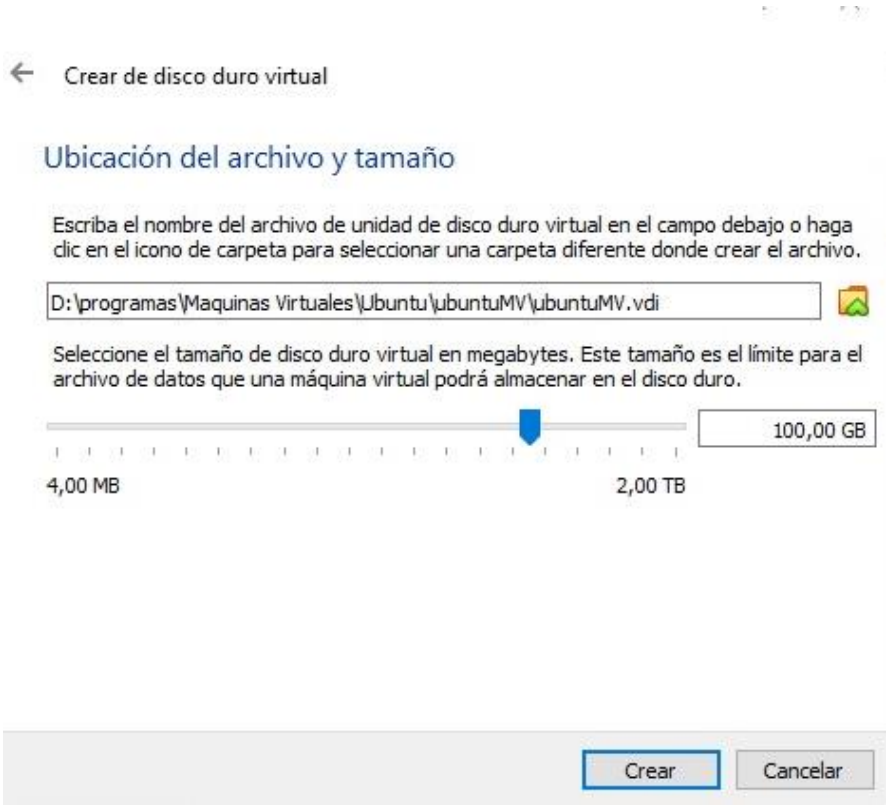
Seleccionamos el tipo de archivo deseamos usar para el nuevo disco duro virtual, para este proyecto se seleccionó la opción VDI (VirtualBox Disk Image)

Ilustración 47 – Tipo de Archivo del Disco Duro

Fuente El autor

Seleccionamos el archivo de destino de la unidad del disco duro virtual y el tamaño del disco duro virtual en megabytes y para finalizar el proceso de instalación oprimimos el botón crear.

Ilustración 48 - Esquema de un sistema de base de datos



Fuente El autor

Desde las interfaces gráficas propias de cada sistema operativo se instala Ubuntu. Para este proyecto se decidió hacer tres copias de esta máquina virtual para posteriormente sean configuradas con cada uno de los tres gestores de bases de datos no relacionales que vamos a manejar (MongoDB, Apache Cassandra y Redis).

Una vez se ha realizado todo el procedimiento para la creación de la máquina virtual el último paso necesario es asignar un disco de instalación a la máquina creada para el primer caso será Ubuntu 20.04 y por otra parte el de Kali.

2) **Ubuntu(20.04):configuración de PHP y MongoDB**

- **Instalación MongoDB en Ubuntu 18.04**

Paso 1: Instalación y configuración de MongoDB

Detalladamente se explicará el proceso de instalación de MongoDB para un servidor de Ubuntu en su versión 18.04, en primer lugar, debemos agregar el repositorio y todos los paquetes que se utilizaran.

Vamos a actualizar la lista de paquetes de aplicaciones, lo podemos hacer con el siguiente comando:

- sudo apt update

El comando que usaremos para agregar el repositorio es:

- sudo apt install -y mongodb

Con este comando. Se instalan varios paquetes que contienen la versión estable más reciente de MongoDB, junto con herramientas de administración útiles para el servidor de MongoDB.

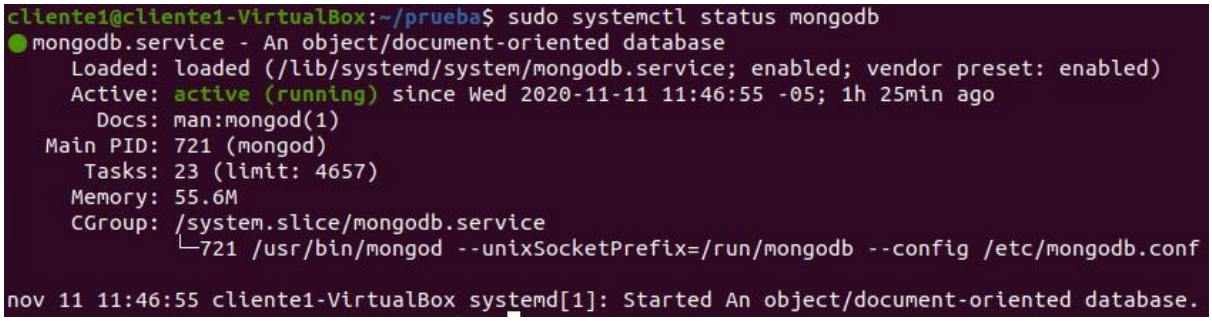
Paso 2: Comprobar el servicio y la base de datos

El servidor de la base de datos se inicia de forma automática tras la instalación, se debe verificar que el servicio se inicia y que la base de datos funcione

- sudo systemctl status mongodb

Si se ejecuta sin errores, este comando generará un resultado similar al siguiente

Ilustración 49 – Status MongoDB



Fuente El autor

Según systemd, el servidor MongoDB se encuentra activo y funcionando.

- **Instalación PHP para MongoDB**

Paso 1: Configuración dentro de las extensiones de MongoDB

Detalladamente se explicará el proceso de instalación del controlador PHP para MongoDB, en primer lugar, debemos agregar el repositorio PHP y todos los paquetes que se utilizaran.

Nota: Si no tienes instalado PEAR y el paquete php-dev deberás instalarlos con

- sudo apt-get install -y php-pear
- sudo apt-get install -y php-dev

Se debe instalar los componentes necesarios como la extensión MongoDB para PHP en tu equipo a través del repositorio de extensiones PHP PECL.

Para descargar e instalar la extensión lo haremos con:

- sudo pecl install mongodb

Esto hará que se descargue el driver, se compile y se instale en nuestro sistema como una extensión.

A continuación, hay que modificar el archivo php.ini para indicar que cargue la extensión de MongoDB (mongodb.so). php.ini se encuentra en:

- /etc/php/7.0/apache2/php.ini.

Ya dentro presionamos Ctrl+W para buscar extensiones dinámicas

NOTE: Ctrl+W, search for "Dynamic Extensions"

Una vez dentro nos desplazamos hasta el final deberás pegar la siguiente extensión en el final de la zona de extensiones

NOTE: add extension=mongodb.so

Ahora, reiniciamos el servidor Apache para que tengan efecto los cambios

- sudo service apache2 restart

Paso 2: Instalación y configuración de La librería PHP

Instalaremos la Librería PHP para MongoDB mediante Composer ejecutando el comando siguiente sobre la raíz de nuestro proyecto

- composer require mongodb/mongodb

De manera automática esto instalará la librería en nuestro proyecto y generará los archivos de autoload.

Para usar la librería en el proyecto comenzaremos con el siguiente script

- <?php

  require 'vendor/autoload.php';

  ?>

3) **Ubuntu (20.04): configuración de PHP y Redis**

- **Instalación Redis en Ubuntu 18.04**

Paso 1: Instalación y configuración de Redis

Detalladamente se explicará el proceso de instalación de Redis para un servidor de Ubuntu en su versión 18.04, en primer lugar, debemos agregar el repositorio y todos los paquetes que se utilizaran.

Vamos a actualizar las listas de paquetes de aplicaciones, lo podemos hacer con el siguiente comando:

- sudo apt update

El comando que usaremos para agregar el repositorio es:

- sudo apt install redis-server

Esto finalizará el proceso de instalación de Redis.

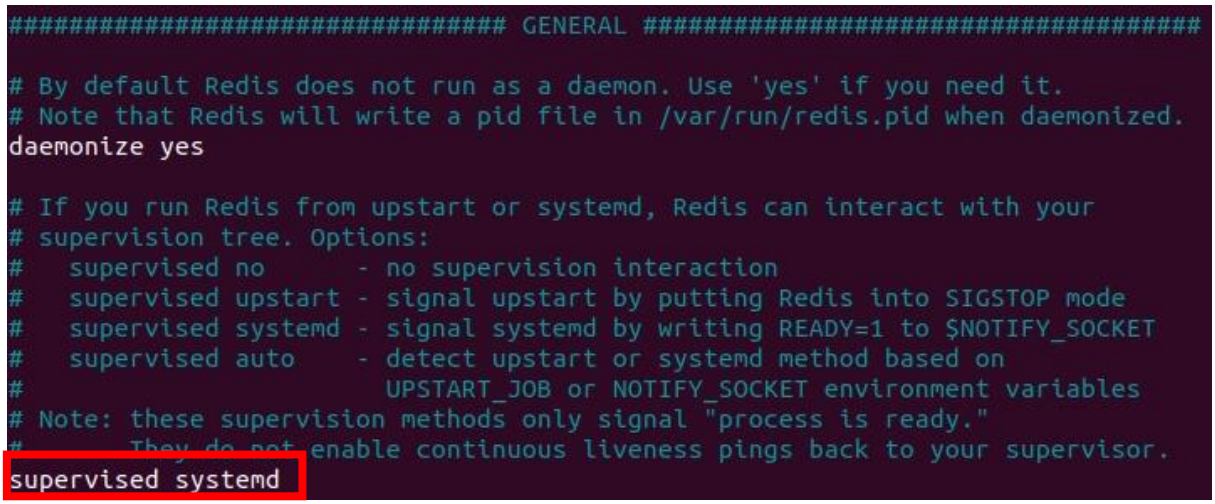
Después de esto, hay un cambio de configuración importante que se debe realizar en el archivo de configuración de Redis, generado automáticamente durante la instalación.

Abre este archivo desde la terminal

- sudo nano /etc/redis/redis.conf

Encuentre la directiva supervised dentro del archivo. Esta directiva le permite declarar un sistema init para administrar Redis como un servicio, lo que le proporcionará mayor control sobre su funcionamiento. Por defecto, el valor de la directiva supervised es no. Debido a que se trata de Ubuntu, el cual utiliza el sistema init de systemd, cambie el valor a systemd como se muestra en la imagen:

Ilustración 50 - Directiva supervised



Fuente El autor

Después de haber guardado los cambios, a continuación, reinicie el servicio de Redis para reflejar los cambios realizados en el archivo de configuración:

- sudo systemctl restart redis.service

Con esto, Redis quedará instalado y configurado, antes de comenzar a utilizarlo, es conveniente verificar primero si funciona correctamente.

Paso 2: Prueba de ejecución en Redis

Primero verificando que el servicio de Redis esté en ejecución:

- sudo systemctl status redis

Si se ejecuta sin errores, este comando generará un resultado similar al siguiente

Ilustración 51 – Status Redis

Fuente El autor

- **Instalación PHP para Redis**

Paso 3: Instalación de la extensión PHP

A continuación, se muestra el proceso de instalación de la extensión PHP de Redis en el servidor web.

Actualice el índice del paquete local e instale el software en su servidor web escribiendo:

- sudo apt-get update

Usaremos la extensión PHP para almacenar los datos de nuestra sesión.

- sudo apt install php-redis

Y reinicie el servidor Apache:

- sudo service apache2 restart

4) Ubuntu (20.04): configuración de PHP y Cassandra

- **Instalación PHP para Cassandra**

Paso 1: Instalación y configuración de La librería PHP

Detalladamente se explicará el proceso de instalación del controlador PHP para Cassandra, en primer lugar, debemos agregar el repositorio PHP y todos los paquetes que se utilizaran.

El comando que usaremos para agregar el repositorio es:

- sudo apt-add-repository ppa:ondrej/php

Luego ingresamos nuestra contraseña de administrador. Una vez se ha terminado de agregar el repositorio vamos a actualizar las listas de paquetes de aplicaciones, lo podemos hacer con el siguiente comando:

- sudo apt update

Paso 2: Instalación de las dependencias

Le instalamos las dependencias para php usando el siguiente comando:

- sudo apt install apache2 php7.1 php7.1-xml php7.1-dev libgmp-dev libpcre3-dev g++ make cmake libssl-dev openssl

Ahora instalamos las dependencias para la Cassandra real.

- wget https://downloads.datastax.com/cpp-dr...
- wget https://downloads.datastax.com/cpp-dr...
- wget http://security.ubuntu.com/ubuntu/poo...
- wget https://downloads.datastax.com/cpp-dr...
- wget https://downloads.datastax.com/cpp-dr...

También se deben instalar estas dependencias, pero deben ser instaladas en el orden exacto como se muestra a continuación:

1. sudo dpkg -i multiarch-support_2.27-3ubuntu1.2_amd64.deb
2. sudo dpkg -i cassandra-cpp-driver_2.15.3-1_amd64.deb
3. sudo dpkg -i cassandra-cpp-driver-dev_2.15.3-1_amd64.deb
4. sudo dpkg -i libuv1_1.35.0-1_amd64.deb
5. sudo dpkg -i libuv1-dev_1.35.0-1_amd64.deb

Ahora que tenemos todas las dependencias que necesitamos instaladas, finalizando el proceso de manera automática se debe usar:

- sudo pecl install cassandra

Para abrir o editar el instalador de Cassandra usaremos:

- sudo nano /etc/php/7.1/apache2/php.ini

Ya dentro presionamos Ctrl+W para buscar extensiones dinámicas

- NOTE: Ctrl+W, search for "Dynamic Extensions"

Una vez dentro nos desplazamos hasta el final deberás pegar la siguiente extensión en el final del texto de la siguiente imagen.

- NOTE: add extension=cassandra.so to end of dynamic extensions section

Ilustración 52 – Pestaña de Extensiones Dinámicas

```
; The MIBS data available in the PHP distribution must be installed.
; See http://www.php.net/manual/en/snmp.installation.php
;extension=php_snmp.dll

;extension=php_soap.dll
;extension=php_sockets.dll
;extension=php_sqlite3.dll
;extension=php_tidy.dll
;extension=php_xmlrpc.dll
;extension=php_xsl.dll
```

Fuente El autor

Para finalizar el proceso presionamos Ctrl+x, luego (Y) y por último Enter.

Paso 3: Reinicio del servidor

Bien, ahora que ha editado la configuración se debe volver a cargarla escribiendo el siguiente comando.

- sudo service apache2 restart

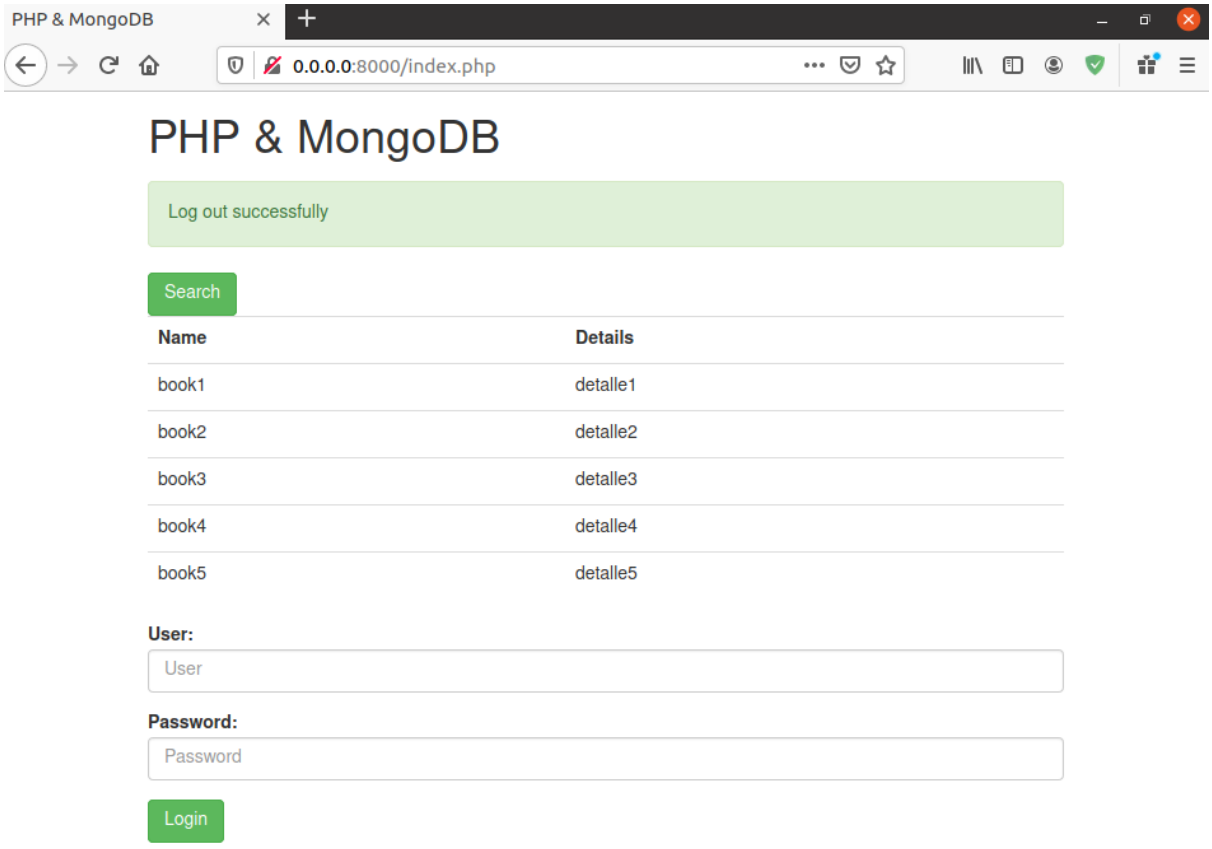
Anexo # 2 – Prototipo

Para el desarrollo del prototipo se escogió el lenguaje PHP como herramienta de desarrollo. Todo el código utilizado para el funcionamiento de los tres prototipos se encuentra en GitHub específicamente en https://github.com/ric-gon/php-mongodb, https://github.com/ric-gon/php-cassandra y https://github.com/ric-gon/php-redis. A Continuación, se explicará de manera detallada el funcionamiento del prototipo.

Índex.PHP

La pantalla principal contiene la lista de todos los elementos en la base de datos, un botón para realizar búsquedas (en donde se ejecutan las pruebas de inyección) y un botón para iniciar sesión cuando se registra el usuario y la contraseña correcta.

Ilustración 53 – Index - Prototipo

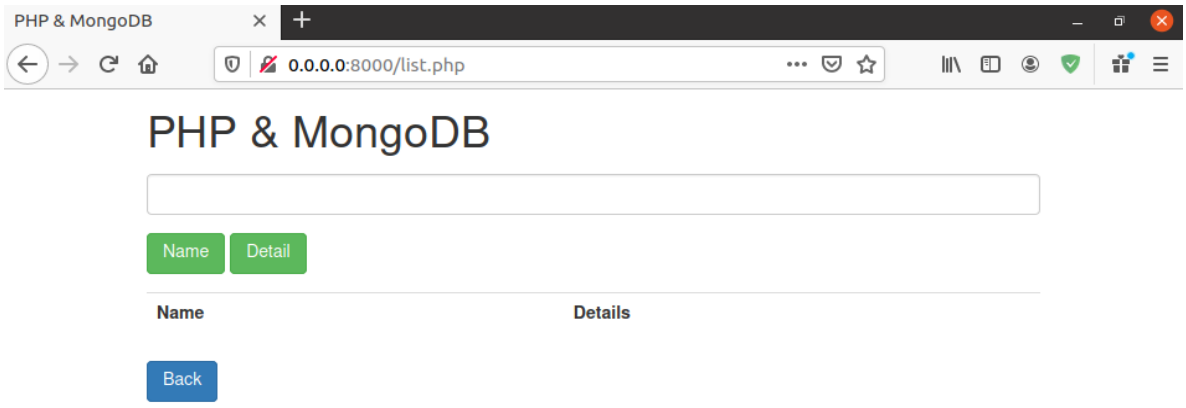


Fuente El autor

Pestaña de búsqueda

En la pantalla de búsqueda en la parte superior se puede ver una barra de búsqueda, un botón que selecciona cual filtro se desea utilizar para realizar la búsqueda y un botón para volver a la pantalla inicial.

Ilustración 54 – Pestaña de Búsqueda- Prototipo

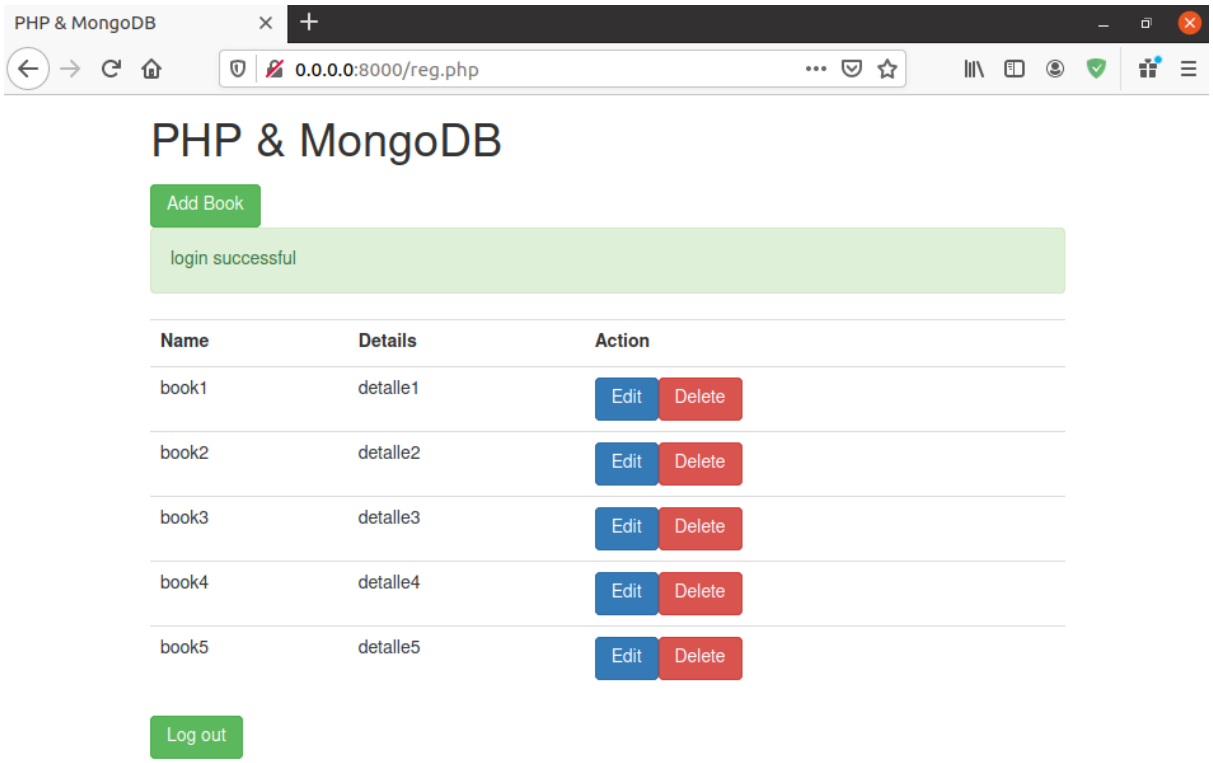


Fuente El autor

Pestaña Administrativa

Cuando se inicia sesión la pantalla muestra al usuario todos los elementos de la base de datos listados cada uno con dos botones uno de eliminar y de editar, en la parte superior se encuentra un botón para agregar elementos a la base de datos y en la parte inferior un botón donde se cierra sesión y vuelve a la pantalla principal.

Ilustración 55 – Pestaña Administrativa - Prototipo

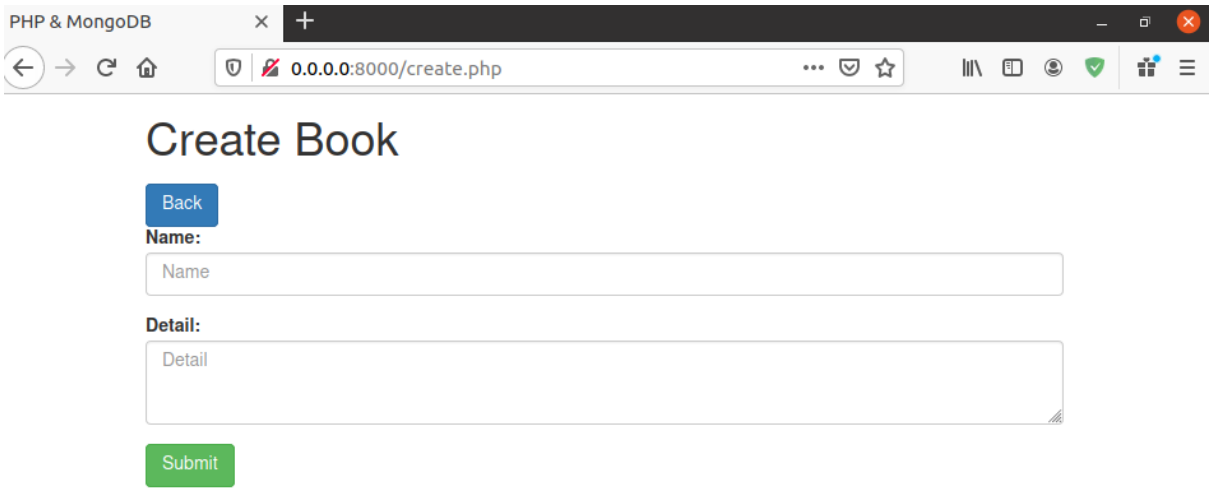


Fuente El autor

Pestaña para Ingreso de Datos

En la pantalla de crear nuevo libro para el ejemplo dado se tiene la opción de cancelar en el botón back y se debe agregar un nombre y los detalles del elemento dependiendo de la base de datos puede requerir agregar una clave y finalmente el botón para guardar los cambios.

Ilustración 56 – Pestaña para Ingreso de Datos - *Prototipo*

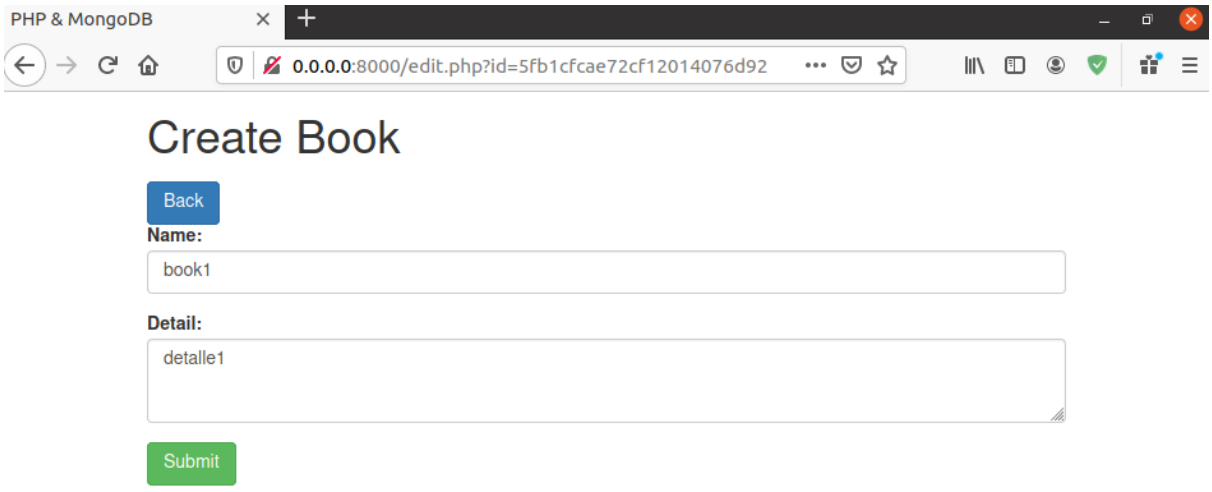


Fuente El autor

Pestaña de edición

La siguiente vista para modificar la información también cuenta con un botón para cancelar y otro para guardar los cambios sin embargo ahora el prototipo muestra la información de un registro en la base de datos.

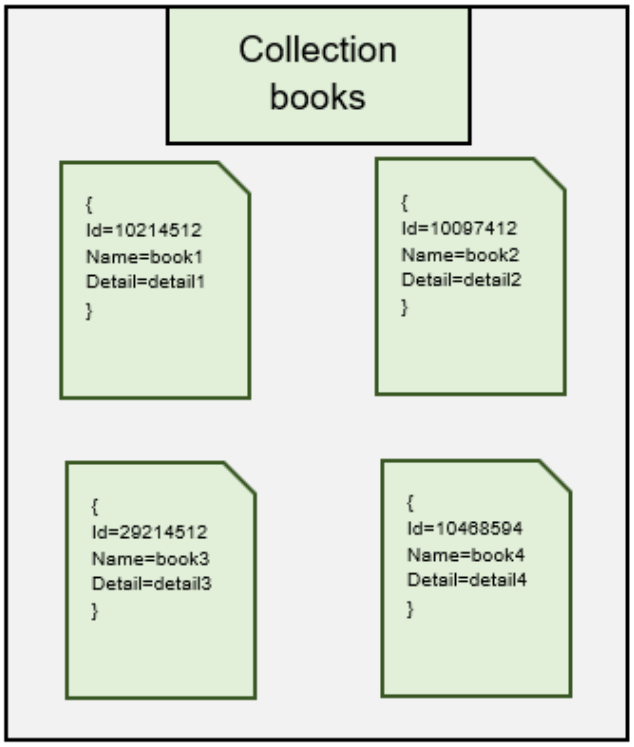
Ilustración 57 - Pestaña de edición - *Prototipo*



Fuente El autor

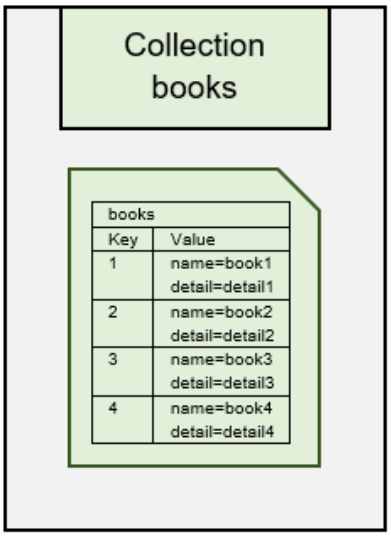
Diagramas de bases de datos NoSQL

Grafica 3 – Document Model



Fuente El autor

Grafica 4 – Key Value



Fuente El autor