Universidad Gerardo Barrios

San miguel

Facultad de ciencia y tecnología

Carrera:

Ingeniería de software y redes informáticas

Materia:

Administración de Base de Datos II

Catedrática:

Gisela Yasmín García Espinoza

Estudiantes:

Chávez Herrera Cesar André.	SMIS033320
Villatoro Pérez Ángel Armando.	SMIS924020
William Rene Jiménez Guzmán.	SMIS031920

Ciclo:

1-2022

Tema:

Analisis de Amenazas, riesgos y vulnerabilidades de las bases de datos

Introducción

Las bases de datos tanto SQL como NoSQL, son dos categorías en los cuales se pueden almacenar los datos dependiendo de las necesidades y en qué proyecto lo vamos a aplicar. Dependiendo de que categoría de bases de datos usemos, así serán los beneficios que nos otorgan, pero en la mayoría de los casos, se elige el tipo de gestor de bases de datos de acuerdo con las necesidades, pero, así como hay ventajas en su uso, pero así hay riesgos, vulnerabilidad y peligros al querer hacer uso de las ya mencionadas, que a continuación expondremos.

Problemas identificados

Amenazas, riesgos y vulnerabilidades de bases de datos SQL.

Gestión de permisos inadecuada

Más a menudo de lo que nos gustaría admitir, los servidores de bases de datos se instalan en organizaciones con su configuración de seguridad predeterminada y esta configuración nunca se cambia. Esto hace que las bases de datos queden expuestas a atacantes que conocen los permisos predeterminados y saben cómo explotarlos.

Ataques de inyección de base de datos

La principal forma de ataques de inyección de base de datos es inyección SQL ataques, que atacan servidores de bases de datos relacionales (RDBMS) que utilizan lenguaje SQL. Las bases de datos NoSQL, como MongoDB, RavenDB o Couchbase, son inmunes a los ataques de inyección SQL pero son susceptibles a los ataques de inyección NoSQL. Los ataques de inyección NoSQL son menos comunes, pero igualmente peligrosos.

Ataques de inyección SQL, implican a un usuario que se aprovecha de vulnerabilidades en aplicaciones web y procedimientos almacenados, para proceder a enviar consultas de bases de datos no autorizadas, a menudo con privilegios elevados.

Vulnerabilidades de bases de datos explotables

Es común que los departamentos de TI corporativos no apliquen parches a su software principal de DBMS con regularidad. Por lo tanto, incluso si se descubre una vulnerabilidad y el proveedor lanza un parche para eliminarla, pueden pasar meses antes de que las empresas parcheen sus sistemas. El resultado es que las vulnerabilidades quedan expuestas durante largos períodos, lo que puede ser aprovechado por los ciberdelincuentes.

Existencia de servidores de bases de datos ocultos

El incumplimiento de las políticas de instalación de software en una organización (o la falta de tales políticas) hace que los usuarios instalen servidores de bases de datos a su discreción para resolver necesidades particulares. El resultado es que aparecen servidores en la red de la organización, algo que los administradores de seguridad desconocen. Estos servidores exponen datos confidenciales a la organización o exponen vulnerabilidades que los atacantes pueden aprovechar.

Copias de seguridad accesibles

Aunque los servidores de bases de datos están protegidos por una capa de seguridad, los usuarios sin privilegios pueden acceder a las copias de seguridad de estas bases de datos. En tal situación, existe el riesgo de que usuarios no autorizados puedan hacer copias de las copias de seguridad y montarlas en sus propios servidores para extraer la información confidencial que contienen.

Falta de contingencia

Las empresas que se jactan de ser “ágiles” y “receptivas”, a menudo alcanzan esa velocidad mediante el abandono de la estandarización, los procesos maduros y el planeamiento de contingencias. Muchas PyMEs descubrieron que un simple fallo o compromiso de los datos se convierte en un desastre cuando no hay Plan de Continuidad de Negocios, Plan de Recuperación ante Desastres, Política de Respuesta ante Intrusiones, sistema de respaldo actualizado desde el cual realmente se pueda hacer una recuperación o almacenamiento en otra ubicación.

Correo electrónico HTML malicioso

El ataque más común por correo electrónico ahora viene como un mensaje en HTML que contiene un enlace hacia un sitio malicioso con alguna trampa caza bobos. Un clic equivocado puede desencadenar una descarga peligrosa. Los riesgos son los mismos que en la Amenaza # 3, "Navegación web imprudente”, pero el atacante utiliza el correo electrónico para llevar a la víctima hacia su sitio web malicioso.

Uso temerario de redes de hoteles y quioscos

Las redes de los hoteles están notoriamente infectadas con virus, gusanos, spyware y malware y, a menudo, funcionan con malas prácticas globales de seguridad. Los quioscos públicos son un lugar conveniente para que un atacante deje un keylogger, sólo para ver qué cae en su red. Las laptops que no tengan software de firewall personal, antivirus y antispyware pueden verse comprometidas cuando están de viaje. Las defensas tradicionales pueden volverse inútiles cuando el usuario, literalmente, transporta la laptop a lo largo del gateway firewall y se conecta desde el interior de la zona confiable.

Riesgos y vulnerabilidades de las bases de datos NoSQL

1. Atomicidad: Algunas de estas bases de datos no incorporan la característica de la atomicidad de información. Esto puede derivar en que la información no sea consistente entre nodos, pudiendo generar algunos problemas en los criterios de análisis.

2. Software poco documentado: Al ser tan relativamente nuevo, el NoSQL puede adolecer de que algunas operaciones sean limitadas por la falta de información sobre las herramientas y sus características. Esto puede ocasionar significativas inversiones de tiempo y dinero para quienes no tienen conocimientos profundos en el área.

3. Baja estandarización: No se tiene un criterio plenamente definido entre los motores que se utilizan en este tipo de base de datos. El lenguaje tiende a variar según el tipo de base de datos que se vaya a utilizar.

4. Herramientas GUI: la mayoría de las bases de datos NoSQL no contienen una interfaz gráfica para el apoyo de herramientas. Se requiere conocimiento especial para poder ejecutar algunas de ellas, lo que limita en gran medida a quienes están iniciándose en este mundo.

Tomado de: "Analysis Of NoSQL Database Vulnerabilities"

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 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.

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:

1. 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.

2. 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
3. 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.

4. 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.

5. 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

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

1. 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.

2. 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.
3. 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.

4. 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.

5. 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 propenso a ataques de pentesting.

Solución los problemas.

Para brindar una protección adecuada a las bases de datos de una organización, se necesita una matriz defensiva de mejores prácticas, combinada con controles internos regulares. La matriz de mejores prácticas incluye los siguientes elementos:

- Administre los derechos de acceso de los usuarios y elimine los privilegios excesivos y los usuarios inactivos.
- Capacite a los empleados en técnicas de mitigación de riesgos, incluido el reconocimiento de amenazas cibernéticas comunes, como ataques de phishing, las mejores prácticas en torno al uso de Internet y correo electrónico, y gestión de contraseñas.
- Evalúe las vulnerabilidades de la base de datos, identifique los puntos finales comprometidos y clasifique los datos confidenciales.
- Supervise toda la actividad de acceso a la base de datos y los patrones de uso en tiempo real para detectar fugas de datos, SQL no autorizado y Big Data transacciones y ataques de protocolo/sistema.
- Automatice la auditoría con una plataforma de auditoría y protección de bases de datos.
- Bloquee las solicitudes web maliciosas.
- Archivar datos externos, cifrar bases de datos y enmascare los campos de la base de datos para ocultar información confidencial.

También es posible utilizar herramientas para facilitar este trabajo.

- Scuba Database Vulnerability Scanner
- dbWatch Control Center.
- AppDetectivePRO.
- DbDefence.
- OScanner.
- dbForge Security Manager.

Basado en una investigación sobre la evaluación de seguridad de las bases NoSQL se pueden tomar en cuenta una gran cantidad de herramientas de análisis de seguridad para realizar escaneos de vulnerabilidades:

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

conclusiones

- Podemos concluir que a pesar de que todas las bases de datos pueden ser alteradas, siempre hay métodos para contrarrestar o disminuir la frecuencia o peligrosidad de dichos ataques.
- Los errores mas comunes que se encuentran en las bases de datos no siempre son culpa de la computadora, los errores humanos pueden generar errores innecesarios y que pueden afectar a la empresa o al servicio en cuestión.
- Las bases Sql tienen un mayor soporte, una mayor comunidad y aplicaciones, esto mismo la hace mas propensa a ser atacadas.
- Las bases Sql Disponen de herramientas que permiten evitar la duplicidad de registros, garantizando la integridad referencial.
- Como las bases de datos NoSQL tienen menos restricciones en relaciones y chequeos de consistencia, son más vulnerables a ataques de inyección, sin embargo, el atacante debe ser experto en programación y sintaxis del lenguaje atacado.