Clase asincrónica 28/05

Bases de Datos II

Luis Diego Delgado Muñoz

Prof. Nereo Campos

Fecha de Entrega: 30/05/2024

Problemas de Seguridad que Afectan a Cassandra DB

1. **Cifrado de Datos**:
    - Cassandra no cifra los datos en reposo. Esto significa que cualquier atacante con acceso al sistema de archivos puede leer los datos directamente de los archivos. No hay un mecanismo incorporado para cifrar automáticamente los datos almacenados en Cassandra. Además, las comunicaciones entre los clientes y los servidores de Cassandra no están cifradas por defecto. Las credenciales de inicio de sesión (nombre de usuario y contraseña) se envían en texto plano y la comunicación entre pods no es cifrada, lo que las hace susceptibles a la interceptación por parte de atacantes que monitorean el tráfico de la red.
2. **Vulnerabilidades del Lenguaje de Consulta**:
    - El Lenguaje de Consulta de Cassandra (CQL) es vulnerable a ataques de inyección. La sintaxis de CQL es similar a SQL, y si los parámetros de entrada no se sanitizan adecuadamente, puede llevar a vulnerabilidades de inyección de SQL.

Problemas de Seguridad que Afectan a MongoDB

1. **Cifrado de Datos y Autenticación**:
    - MongoDB no proporciona cifrado incorporado para datos en reposo. Los datos sensibles son vulnerables si un atacante obtiene acceso al sistema de archivos subyacente. También, los mecanismos de autenticación de MongoDB son limitados, especialmente en configuraciones fragmentadas. Solo las conexiones nativas admiten la autenticación básica, y esto solo está disponible en configuraciones no fragmentadas. Además, las contraseñas se almacenan utilizando hashes MD5, que no se consideran seguros.
2. **Autorización**:
    - La autorización en MongoDB es rudimentaria, proporcionando solo roles de lectura y lectura-escritura sin soporte para el control de acceso granular. En configuraciones fragmentadas, no hay soporte para autenticación y, por lo tanto, no hay autorización. Los permisos granulares pueden gestionarse externamente usando un proxy inverso.
3. **Ataques de Inyección**:
    - MongoDB es vulnerable a ataques de inyección a través de su uso de JavaScript y concatenación de cadenas en consultas. La validación de

entrada es necesaria para prevenir estos ataques.

Mejorando la Seguridad en Cassandra y MongoDB

Para mejorar la seguridad en estos sistemas de BD yo implementaría cifrado a nivel del sistema de archivos para proteger los datos en reposo y también usar cifrado a nivel de aplicación para cifrar datos sensibles antes de que se almacenen en la base de datos. Además, pensaría en habilitar SSL/TLS para todas las comunicaciones cliente-servidor y entre clústeres para proteger los datos en tránsito de ser interceptados por atacantes.

En ambos sistemas sería sumamente importante fortalecer los mecanismos de autenticación integrándose con protocolos de autenticación más seguros y definitivamente evitar el uso de algoritmos de hash débiles como MD5 para almacenar contraseñas.

Para prevenir ataques de inyección, se implementaría una validación y sanitización adecuadas de las entradas para prevenir ataques de inyección, además de usar declaraciones preparadas o consultas parametrizadas para evitar la inclusión directa de entradas de usuario en las consultas.

Al abordar estos problemas de seguridad, tanto Cassandra como MongoDB pueden ser significativamente reforzados contra varios tipos de ataques y accesos no autorizados.