**Security Analysis of MongoDB – A Literature Review**

Due to recent advancements in cloud computing platforms and the emergence of Big Data, NoSQL Databases are more becoming popular than ever. NoSQL Databases are a form of non-relational databases whose primary purpose is to store and retrieve data. (Singh, 2019)

**Introduction**

NoSQL models are generally quicker and are able to process large amounts of heterogenous data compared to relational database models hence often being the first choice while working with large and unstructured datasets owing to their speed and flexibility. (Singh, 2019)

**Features of MongoDB**

MongoDB is a document-based database developed by 10gen which manages collection of JSON like documents format called BSON or simply Binary JSON”.

* Map reduce based Aggregation Framework: Map reduce based Aggregation Framework: This feature of MongoDB is similar to the ‘Group By’ clause offered in MySQL. A map is basically a procedure for filtering and sorting data while reduce procedure performs a summary operation (Eg counting the number of people standing in a queue). (Singh, 2019)
* Schema Less Database: By schema-less we are referring to dynamically typed schema as opposed to statically typed schemas in Relational Databases. Since there is no constraint on the data and every document in the collection can have different attributes from each other we call it schema less. In the case of relational databases such schema is defined using tables. The schema refers to the structure in which the data should be stored. (Singh, 2019)
* Ad-hoc Querying: MongoDB supports SQL like complex queries including regex. (Singh, 2019)
* Replication and fail-over support: MongoDB supports replication by distributing data over various clusters, this is achieved using replica set which is essentially a group of instances hosting the same data. All write operations are assigned to the primary node (i.e. the master node) whereas the secondary nodes may perform read operations. (Singh, 2019)

In a replica set, one node is defined as the primary node while all other nodes classify as secondary.

**Security Features**

* Enabling Access Control: MongoDB requires that all users (clients as well as servers) provide valid credentials before they are able to connect to the system. With Access Control enabled, there is a user administrator role defined which is responsible for creating users, granting and revoking access of other users as well as modifying user roles. Enabling access control enforces authentication and requires all users to identify themselves before a connection is made. (Singh, 2019)

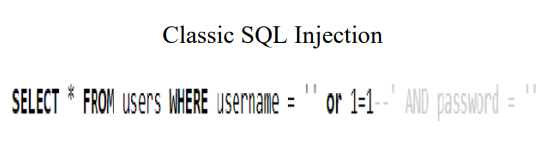


After the admin has been created, you can then go on to create additional users based on exact access principle (giving the least privilege required). (Singh, 2019)

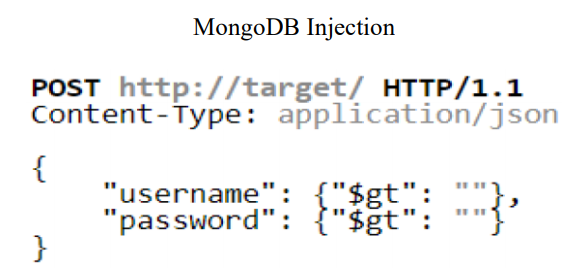
* Confidential Network: MongoDB only allows users to connect over defined interfaces on a given port in which MongoDB instances are available.
* IP Binding: Starting MongoDB 3.6, MongoDB binaries bind to localhost by default.
* Network Hardening: One way to achieve this is using Firewalls which limits traffic to only those from trusted sources. The other way is to use Virtual Private Networks which makes it possible to two networks over an encrypted and limited access network.

**Security Flaws and Addressal**

* Lack of Data Encryption: Currently there is no encryption on data files in MongoDB. This is a cause of concern since anyone with access to file systems can extract the information from these files. (Singh, 2019)
* Vulnerable to Injection Attacks: Simply because MongoDB does not deal directly with a query language in the form of string does not make it immune to injection attacks (See example below on how injection attacks take place in SQL). (Singh, 2019)

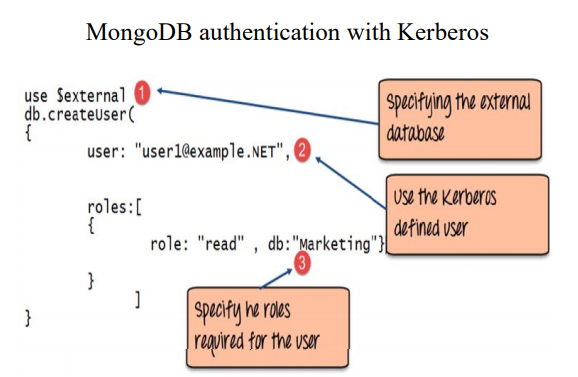


The above example shows how SQL injections take place since 1=1 is always true, the above query will be executed, and attackers might be able to get access to private information (E.g. the above table might contain usernames and passwords.) Hence such statements might be unknowingly be running on your database



In the above example, username and password have not been validated to ensure they are strings hence they may contain any field but strings and manipulate the query structure.

* Authentication and Authorization: Any user by default has the permission to access the entire database, moreover any user with administrator access has complete read/write privileges for the complete database. MongoDB does not provide authentication in sharded configuration unless run in standalone or replica set mode also the onus of security lies entirely in the hand of the developer. (Singh, 2019)



**Mysql Vs Mongodb Security Comparsion**

* Security Model: MySQL provides a privilege-based security model i.e. providing a user which access to only specific commands such as CREATE, UPDATE, DELETE etc. (Singh, 2019)
* Injections: MySQL is prone to SQL injections which is essentially placing malicious code in SQL statements via web page output. (Singh, 2019)
* Logging: MySQL offers complete logging by default and supporting transaction and rollbacks helps in ensuring data integrity. (Singh, 2019)
* Integrity Model: MySQL follows ACID (Atomic, Consistent, Isolated, Durable) model. Database administrators use several strategies to enforce ACID such as write ahead logging (WAL), shadow paging and two-phase commit protocol. MongoDB follows the BASE (Basic Availability, Soft state, Eventual consistency) model. (Singh, 2019)

**Data Breach Cases**

1. The data comprised of three folders with about 800 million records in one (emailrecords), 4 million contact information record in another (emailWithPhone) and around 6 million records of business leads in the third folder (businessLeads) which included information such as mortgage details and other corporate information. (Singh, 2019)
2. • Security researchers discovered over 808 million records including sensitive information such as contact number and email address being exposed on a MongoDB instance.
3. A security researcher named “Bob Diachenko” exposed a vulnerability in which was leaking the details of around 11 million users belonging to an email marketing firm based out of California. (Singh, 2019)
4. The dataset contained around 44 Gigabytes of data including full names, email addresses, gender details and physical addresses of 10,999,535 users. (Singh, 2019)
5. Each of the 200 million resumes also contained personal information such as contact number, email address, height, weight, driver license, salary expectations etc. (Singh, 2019)

**Common Vulnerabilities and Exposures**

1. MongoDB earlier versions (between 2.10 and 2.20) do not properly validate requests to native helper function in SpiderMonkey allowing remote authenticated users to cause a denial of service attack or executing arbitrary code through a crafted memory address in the first segment. (Singh, 2019)
2. The MongoDB Js-Bson module versions before 1.0.5 are vulnerable to Regular expression Denial of Service (ReDoS), the flaw being triggered on calling fromString() function to parse a long untrusted string. (Singh, 2019)
3. Certain MongoDB versions between 2.4 and 2.6 provide vulnerabilities in security to allow a denial of service attack using UTF-8 string in a BSON based request. (Singh, 2019)

**Conclusion**

MongoDB on the other hand does not offer most security configurations by default however, it’s still less prone to injections attacks considering it does not directly deal with a query language in the form of string. Having the flexibility for development without any predefined schema is a massive boost for MongoDB however it is essential that it is backed by a robust security mechanism to prevent malicious attacks and unwanted interferences. On the other hand, if you have unstructured data or an undefined schema in hand and want to process large amounts of data MongoDB would be a reasonable choice. (Singh, 2019)

# Bibliography

Singh, S. (2019). *Security Analysis of MongoDB*.

Related Links- https://www.researchgate.net/publication/343224438\_Security\_Analysis\_of\_MongoDB