Asian Journal of Research in Computer Science

14(1): 1-12, 2022; Article no.AJRCOS.87476
ISSN: 2581-8260

Analysis of the Unexplored Security Issues Common to All Types of NoSQL Databases

Hima Bindu Sadashiva Reddy [a*], Roopesh Reddy Sadashiva Reddy [b], Ratnaditya Jonnalagadda [c], Pallavi Singh [d] and Avinash Gogineni [eφ]

[a] College of Computing and Engineering, Nova South Eastern University, Davie, Florida, USA.
[b] Oklahoma State University, StillWater, Oklahoma, USA.
[c] Wharton Business School, University of Pennsylvania, Philadelphia, PA, USA.
[d] Steve Hicks School of Social Work, The University of Texas at Austin, Austin, TX, USA.
[e] University of Miami, Miami, Florida, USA.

Authors' contributions

This work was carried out in collaboration among all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJRCOS/2022/v14i130323

Open Peer Review History:
This journal follows the Advanced Open Peer Review policy. Identity of the Reviewers, Editor(s) and additional Reviewers, peer review comments, different versions of the manuscript, comments of the editors, etc are available here: https://www.sdiarticle5.com/review-history/87476

Original Research Article

Received 12 March 2022
Accepted 23 May 2022
Published 24 May 2022

ABSTRACT

NoSQL databases outperform the traditional RDBMS due to their faster retrieval of large volumes of data, scalability, and high performance. The need for these databases has been increasing in recent years because data collection is growing tremendously. Structured, unstructured, and semi- structured data storage is allowed in NoSQL, which is not possible in a traditional database. NoSQL needs to compensate with its security feature for its amazing functionalities of faster data access and large data storage. The main concern exists in sensitive information stored in the data. The need to protect this sensitive data is crucial for confidentiality and privacy problems. To understand the severity of preserving sensitive data, recognizing the security issues is important. These security issues, if not resolved, will cause data loss, unauthorized access, database crashes by hackers, and security breaches. This paper investigates the security issues common to the top twenty NoSQL databases of the following types: document, key-value, column, graph, object-oriented, and multi-model. The top twenty NoSQL databases studied were MongoDB, Cassandra, CouchDB, Hypertable, Redis, Riak, Neo4j, Hadoop HBase, Couchbase, MemcacheDB, RavenDB, Voldemort, Perst, HyperGraphDB, NeoDatis, MyOODB, OrientDB, Apache Drill, Amazon, and Neptune. The comparison results show that there are common security issues among the

[φ]Clinical Research Coordinator;
**Corresponding author: E-mail: hs664@mynsu.nova.edu;*

databases. SQL injection security issues were detected in eight databases. The names of the databases were MongoDB, Cassandra, CouchDB, Neo4j, Couchbase, RavenDB, OrientDB, and Apache Drill.

Keywords: NoSQL; security issues; document; key-value; column; graph; object-oriented; multi-model databases.

1. INTRODUCTION

Not Only SQL (NoSQL) was originally introduced by Carlo Strozzi, which is an open-source database that stores data in the form of shell scripts and ASCII files in place of SQL [1]. These databases are mainly non-relational database systems which uses BASIC (Basically Available, Soft State, Eventual consistency) properties, whereas traditional RDBMS uses ACID (Atomicity, Consistency, Isolation, Durability) properties [2,3,4]. The most common advantages of the NoSQL databases are faster data reading and writing, large volume data storage, cost-effectiveness, flexible structure and easier expansion [5,6]. They are used in numerous fields including manufacturing, health care, bioinformatics, social media/network and e-commerce. NoSQL was mainly used to address the drawbacks of RDBMS in web applications; the drawbacks were related to performance and scalability [2,4]. Additionally, NoSQL databases handle huge data with high performance, flexibility, and availability [7]. MySQL and MongoDB databases were compared using Yahoo! Cloud Serving Benchmark. The results supported NoSQL MongoDB for its performance. However, the researchers also emphasize that choosing a database is purely based on the parameters such as the size of the environment, read or write performance, extensibility, coherence, redundancy, and replication. The main intention for any organization to select a NoSQL database is for performance [8].

An enterprise's performance solely depends on which NoSQL database has been selected because deciding on a perfect one from a list of the 225 NoSQL databases available is a challenging task [9]. Gessert [10] proposed a strategy to select from the variety of NoSQL databases based on the organizations' requirements. The requirements were classified based on data access for fast lookups, complex querying, data volume, and query pattern. For example, Redis and Memcache single-node databases are more suitable for organizations with a single machine occupying all the data. Whereas organizations looking for traditional RDBMS or graph databases can choose Neo4J. RDBMS and NoSQL databases are mostly compared to guide Big Data and Cloud Computing companies in choosing a best database (Matallah et al., 2017).

Deka [11] performed a thorough analysis of system capability comparisons for fifteen NoSQL databases based on storage types such as column, document, key-value/tuple, data grid cloud, and MySQL. The system capability comparisons contemplated were data-handling techniques and billing practices. Similarly, Okman et al. [12] addressed the main security issues common to two types of NoSQL databases; Cassandra (column type) and MongoDB (document type). The security issues were related to encryption, lack of authorization related to role-based access control (RBAC), SQL injection, poor authentication, and Denial of Service attacks (DOS). However, there has been no evidence or information provided by the research studies that discusses the comparison of security issues common to all types of databases. There has not been enough explanation for not considering all types of NoSQL databases to compare security issues, amidst the security attacks being reported recently in the Common Vulnerabilities and Exposures (CVE) website. This raises a question if there are common security issues among all types of databases or if only Cassandra and MongoDB are having common security issues. Most of the big data platforms by default use NoSQL databases and do not have inbuilt security as implemented in traditional RDBMS [13]. The need for built-in security for NoSQL databases is important to prevent future security attacks; because big data applications using NoSQL databases are prone to severe security issues which may result in the destruction or misuse of any kind of data [14].

This study intends to find and compare the common security issues existing among all types of NoSQL databases. Hou et al. [14] proposed a defense solution and detective mechanism for MongoDB to prevent Javascript and HTML

injection attacks. One interesting observation made was that there is not sufficient proof of whether this solution can be applied to all types of NoSQL databases. Are these attacks occurring in key- value/tuple, data grid cloud, and MySQL type of databases? This research study will help in finding the gap to discover a firm solution common to all types of NoSQL databases. A comparison of the security attacks will be based on real-time issues reported. NoSQL database's demand on high scalability and availability due to storage and data is increasing rapidly [12]. The emphasis should be given to discover different methods to find one common solution for security issues and to implement built-in security for all types of NoSQL databases.

There has been research conducted describing the major functionalities of NoSQL databases. However, very limited research related to security issues of all NoSQL databases has been conducted. The goal of this study is to explore and compare common security issues existing among all types of NoSQL databases. Here, types of NoSQL databases considered are key-value store, document store, column-family, and graph databases [15]. Additionally, object-oriented, and multi-model NoSQL database types are explored. Noiumkar and Chomsiri [16] studied security issues in MongoDB, Cassandra, CouchDB, Hypertable, and Redis. Security issues dating back to the year 2014 were reported for MongoDB, CouchDB, and Cassandra. Researchers mention that there was no security issue reported for Hypertable and Redis. The research was from the year 2014 and covers only five open source databases of categories document, key-value, and column; graph database is missing. Additionally, there is no information given about any security patches, upgrades, fixes, and solutions for the issues reported. This study intends to find the security issues occurring in the top 20 NoSQL databases for the year 2020 and explore if the issues are common to all the databases. Additionally, this research will further analyze to find if the solution or fix was applied to these security issues. To build the common security mechanism, it is necessary to find common security issues existing across all types of NoSQL databases.

Okman et al. [12] explored the reason for Amazon and Google to promote the NoSQL databases, reasons included large data increase, storage increase, provide high availability, and scalability. The advantages of the NoSQL database are extensive market interest and faster retrieval of information to the user. As the architecture for NoSQL is not relational, this poses security risks for sensitive information. The researchers report issues related to encryption, authorization, denial of service attacks, and SQL injections in two commonly used NoSQL databases, MongoDB, and Cassandra. The study by Okman et al. [12] clarified the importance of preventing security attacks because online transactions have enormous sensitive information. Delay in addressing any security issue will lead to loss of sensitive information, security breaches, database crashes by hackers and unforeseen damage to organizations using NoSQL databases. One of the aims of this study is to discover the unexplored common security issues among document, key-value, column- family, and graph database. Application designers, developers, and NoSQL database administrators would be most benefited to implement a solution for inbuilt security features common to the NoSQL database. Finding security issues in this study will even help the customers to make the correct decision to buy a perfect NoSQL database which is of a less threat to their application [16].

2. LITERATURE REVIEW

This study focuses on exploring and analyzing the unexplored security issues common to all types of NoSQL databases. The four main topics identified to establish the viability for exploring the common security issues are document, key-value, column-family, and graph database. Exploring the literature in connection with the research problem stated led to the detecting of diverse research studies which helped to hypothesize the four constructs that are document, key-value, column-family, and graph database (Lin et al., 2016c). Each type of NoSQL database reports its various security issues. Understanding these security issues will help in decision making for customers buying the product, application designers, developers, and database administrators. Key-value, column, and document are the most used NoSQL databases [17].

2.1 Document Database

NoSQL's document data model is considered to be robust and beneficial to represent complex information; Amazon's SimpleDB mechanism is based on long text-based attribute content (document), document serialization, and indexing

with a key [17]. This mechanism is schema-less, which helps in heterogenous semi structured data storage and high flexibility. The reference keys in the document-oriented database are similar to a foreign key in RDBMS [18]. The keys are unique and are associated with each document collection [19].

Hou et al. [14] selected a document database called MongoDB to study the mechanism and its security concerns. Their article explored a type of security attack called preparatory step attack which raised security concerns for the MongoDB NoSQL database. Further, the researchers describe the importance of solving this type of NoSQL database injection by providing defense and detection solutions by understanding the mechanism of MongoDB. Another study by Kumar and Garg, [20] explains that NoSQL document base MongoDB has its own set of vulnerabilities, one of them being unauthorized access to the filesystem and sensitive information data leakage. This security issue was analyzed by using different cryptographic techniques, encryption, and decryption execution time. Tian et al. [21] performed a deep analysis of data encryption in MongoDB to propose a transparent middleware implementation to secure sensitive data. This research throws light on transparent encryption and decryption of dataset level and its storage implementation in MongoDB applications. CouchDB is a document database with document re- distribution, high scalability, and availability [22]. Security features for CouchDB include encryption and CRUD features to authenticate between the client and server [16]. Researchers reported an interesting vulnerability, where the hackers crashed the CouchDB servers by sending just one line of command.

2.2 Key-value Database

Large scale data-intensive application in commercial, academic, e-commerce platforms, picture stores, and web object caching opt for high-performance key-value store NoSQL databases [23]. The advantages of key-value are size, index memory efficiency, and scalability. Different kinds of key-value databases are Redis, Tokyo Cabinet- Tokyo Tyrant, and Flare [5]. Voldemort, DynamoDB, and Hypertable are also key-value databases (Deka, 2014).

Research by Zaki and Indiramma [24] completely concentrates on Redis a key-value type NoSQL database. The database's key features and security drawbacks are also explored. They proposed an algorithm that performs encryption and decryption faster than other algorithms. The solution implemented makes Redis more secure to be adopted in real- time and multimedia related applications. The most famous key-value database is Amazon's DynamoDB known for its scalability and no downtime (Deka, 2014). Müller et al. [25] experimented on DynamoDB and Transport Layer Security (TLS). The results showed no degradation in performance; hence the authors recommend using DynamoDB only with TLS activated.

2.3 Column Database

Column databases are well known for their high performance and business intelligence processing; commonly used column databases are HBase, HadoopDB, Cassandra, Hypertable, Bigtable, and PNUTS [5].

Cassandra is the database implemented by Facebook with features of fault tolerance, high availability, and scalability [26]. The authors cite security concerns of Cassandra as weak password encryption, DOS, and CQL (Cassandra Query Language) injection. HBase is a column database which has its native encryption features built-in. The experiment by Pallas et al. [27] showed degradation in performance when security features enabled for data confidentiality; the cost was 90% for one of the test results.

2.4 Graph Database

The data is represented in the form of graphs, such databases are used by social media networks [15]. The graphs are also extensively used for website link structures, and field of biology for protein, metabolic, gene, genetic, and chemical mapping [28]. Hulburt [29] describes the importance of failing to protect the privacy of personal data, and redundancy in providing accurate information for future predictions (for example weather forecast) would make the highly reliable graph databases worthless. The author gives a classic example of faulty prediction causing the wrong distribution of resources during disaster relief. Research by Di Martino et al. [30] was the only paper that discusses a graph database called InfluxDB which outperforms both Cassandra and MongoDB. Neo4J is a graph database with SSL as its only security feature [26].

2.5 Multi-model Database

Orientdb.org describes the multi-model database as a database management system that supports document, key- value, graph, and object models. Such databases help in speed and scalability as this database operates as one and providing features of all four models. OrientDB is a classic example of this type of database.

2.6 Object-oriented Database

Information collected from neodatis.wikidot.com explain their main objective to develop Neodatis an object-oriented NoSQL database. Here the data retrieval does not use tables, instead relies on objects. Advantages are simple, fast data access and retrieval, reliability, ease of use, easy integration, and multi- platform. The most important benefit is data is always available as it is stored in XML format; this helps in both import and export of the data. MyOODB is another object-oriented database that is among the top 20 NoSQL databases.

Okman et al. [12] compare security issues between two categories of database document (MongoDB) and (Cassandra). Müller [25] covers an important aspect related to neglecting the security mechanisms in the NoSQL systems. DynamoDB, Cassandra, HBase, MongoDB, and Voldemort are studied to propose a solution based on the degree of vulnerability and security attacks. Amazon DynamoDB and an Apache Cassandra were given more coverage to test the solution implemented. Sahafizadeh and Nematbakhsh [26] conducted a survey to study security issues in MongoDB, Cassandra, CouchDB, HBase, HyperTable, Voldemort, Redis, DynamoDB, and Neo4J. However, the survey reported a possibility of the security attacks (script injection and DOS) based on the security features of each category of the databases; no real-time security issues or solutions were discussed. The security features were authentication, authorization, and data encryption. It was observed from the literature review that although there have been newer NoSQL databases researched for security issues, MongoDB and Cassandra were the most commonly studied databases. It can be learned that security issues do exist among different types of NoSQL databases and few previous research studies have compared security issues among the same type of NoSQL databases. However, there is no proof explaining if the issues reported in MongoDB was found in CouchDB, both the databases fall in the category of document database. Hence the need to study common security issues for all types of NoSQL databases is required.

3. METHODOLOGY

3.1 Approach

Grounded Theory Methodology (GTM) is used as the research design. The initial phase of this study involved researching the types of NoSQL databases, selecting the databases for this study, and researching the resources where the security issues for NoSQL are reported. All of this information was gathered from the literature review process. Open coding was used for the initial phase. The second phase of the study utilized focused coding. The second phase involved analyzing the various security issues collected from the resources. Careful observations and comparisons of security issues between the NoSQL databases were followed comprehensively. This approach is taken because unexplored common security issues should be unearthed, which requires focused and detailed analysis. The results acquired from second phase helped decide that the third phase is not required for this study.

3.2 Data Collection

Theoretical sampling method is used in this research study. Top 20 database names were selected from a website called Big Data Analytics News (BDAN) for the year 2020 [16]. The sample for this research study was data retrieved from websites that report the latest NoSQL security issues. The websites are SecurityFocus, exploit-DB, Redhat, CVE, and CVE details [16]. Additional websites for data collection are community.neo4j.com, couchbase.com, couchdb.apache.org, orientdb.com, and issues.apachedb.org. This study was conducted as a contrived study using the researcher's laptop or desktop as an environment in which the subjects were normally studied.

A list of company names for each NoSQL database was obtained from websites called HG Insights (www.discovery.hgdata.com) and StackShare (stackshare.io).

3.3 Research Questions

Research questions for this study are:

- RQ1: Are there common security issues occurring among all types of NoSQL databases?
- RQ2: Are solutions provided for these common security issues?

3.4 Milestone

A total of 40 to 50 days was required to finish the data collection and analysis. The first 10 to 15 days were dedicated to find if manual data collection will work for all the five database types. Manual data collection worked as per schedule, and no additional time was required to write an automated script for data collection. The next 15 days were used to sort and organize the data collected. The last 15 days were used to evaluate the data and collect the necessary additional details.

3.5 Resources

Windows 10 (8u51 and above) 64-bit operating system, high-speed internet, Google Chrome (version 86 & 64 bit), and Microsoft Excel was used for this research.

4. RESULTS AND DISCUSSION

Tables 1 and 2 in the Appendix give a comparison of the types of databases and their security issues. Table 3 gives the list of companies using these NoSQL databases. Below sections a) through f) explain common security issues found with each type of NoSQL database. This will help the developers to find a solution to design better security solutions for each type of database. In the summary section findings of security issues common to all types of databases are described. This will help in designing an in-built common security solution for all types of databases.

4.1 Document Database

Document database reported security issues such as cross- site scripting (XSS) attacks, denial of service (DOS), code execution, bypass something, gain information, gain privileges, cross-site request forgery (CSRF), SSL vulnerability. CouchDB and Couchbase have common security issues such as XSS, DOS, code execution, gain privileges, and gain information. However, MongoDB, CouchDB, and Couchbase have three security issues in common: DOS, code execution, and gain information. From 2017 to 2019 more number of code execution issues in Couchbase and CouchDB was related to Remote Code Execution (RCE) compared to local code execution. Additionally, both MongoDB and CouchDB were facing security issues such as Bypass something and gain information. CSRF and SSL vulnerability issues were only observed in Couchbase.

4.2 Key-value Database

Key-value database reported issues such as man-in-the- middle attack, DOS, undocumented service access, HTTP request smuggling, XML external entity (XXE), code execution, bypass something, gain information, overflow, and memory corruption. Cassandra, Redis, and MemcacheDB were databases with common security issues such as DOS and code execution. Redis and MemcacheDB faced issues such as bypass something and overflow. Whereas Cassandra additionally had security issues such as man-in-the-middle attack, undocumented service access, HTTP request smuggling, XML external entity (XXE) which were not reported by Redis, Riak, MemcacheDB, and Voldemort.

4.3 Column Database

Column databases reported XSS, DOS, gain information, directory traversal and overflow. Surprisingly, there were no common security issues observed between Hypertable and Hadoop Hbase.

4.4 Graph Database

Among the three graph databases Neo4J, HyperGraphDB, and Amazon Neptune, only Neo4J reported security issues. XXE and CSRF are two security issues faced by Neo4J.

4.5 Object-oriented Database

Object-oriented databases such as Perst, Neodatis, and MyOODB did not report any security issues.

4.6 Multi-model Database

OrientDB, a multi-model database, faced security issues such as code execution, gain information, and CSRF.

Apache drill which supports all types of NoSQL databases, reported issues such as XSS and gain information [31,32]. SQL injection vulnerabilities in MongoDB as reported by cve.mitre.org, allows an unauthenticated attacker to authenticate and access data. Remote Command Execution (RCE) allows the hacker to access the complete database server and log files that have sensitive information. The couchbase.com website reports that the latest version of the CouchBase server is prone to this RCE kind of vulnerability. The attacker hacks the credential details stored in the format of "magic cookie" in the server log files. Next, the attacker will have complete control to run operating system level commands.

Code execution is the security issue that was common in four NoSQL database types; document, key-value, graph, and multi-model. Gain information is the security issue common to five types of NoSQL databases; document, key-value, column, multi-model, and Apache Drill (this one database supports all types of NoSQL databases). Denial of Service (DOS) attack is common in three types of NoSQL databases: document, key- value, and column [33,34]. XSS attack is common to three types of NoSQL databases: document, column, and Apache Drill. Most of the key-value databases did not face the SQL injection security issue. The reason a security company named sqreen reports is that Redis uses JSON to query for interpretation instead of SQL. SQL injection issues were common to document, key-value (Cassandra), graph, and multi-model types of databases. Apache Drill also faced SQL injection attacks.

XXE security issue was common to key-value and graph types of databases. Bypass something security issue is common to document and key-value database. Directory traversal security issues were common in document and column type of NoSQL databases. However, only document and key-value have overflow security issues to be common in both the databases. Finally, it was learned that CSRF security issue was common in three types of NoSQL databases: graph, multi- model, and document.

5. CONCLUSION

The demand for NoSQL databases has seen a tremendous increase in recent years. The need for these databases is mainly due to both the customer's requirement for a better user experience and companies fulfilling their requirements by providing loads of information with no wait time (faster data retrieval). All the amazing advantages of NoSQL are susceptible to security issues as well. This poses risk to users with all the sensitive information which is displayed or transacted online. This study analyzed security issues common to six types of NoSQL databases: document, key-value, column, graph, object-oriented, and multi-model. There were security issues common to different types of NoSQL databases. One observation worth mentioning is, five NoSQL database types document, key-value, column, multi-model, and Apache Drill have gain information as a common security issue. Similarly, code execution, DOS, and XSS were common to three to four types of NoSQL databases. Remote code execution issues reported outnumbered local code execution in CouchBase and CouchDB. Apache Drill is the only database in the top 20 databases which supports all types of NoSQL database. Database designers can think of solutions for built-in security for Apache Drill. It will form a "one solution fits all" concept because Apache Drill supports all types of NoSQL databases.

The limitations of this study are only the top 20 NoSQL database types were considered for this study. Few famous databases commonly used by Google, and Amazon were studied. Data were manually collected from the vulnerability websites. A detailed list of solutions for the security issues was not gathered.

Future work will include all 225 NoSQL databases, provided a script is written to avoid the manual collection of data. If the security fixes applied to these security issues are common to all types of NoSQL databases has scope for further investigation.

ACKNOWLEDGEMENT

The authors would like to thank professor Dr. Ajoy Kumar and fellow students at Nova Southeastern University in guiding and helping with this research study.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Berg KL, Seymour T, Goel R. History of databases. International Journal of

Management & Information Systems (IJMIS). 2013;17(1):29-36.
2. Chandra DG. BASE analysis of NoSQL database. Future Generation Computer Systems. 2015;52:13-21.
3. De Oliveira VF, Pessoa MADO, Junqueira F, Miyagi PE. SQL and NoSQL Databases in the Context of Industry 4.0. Machines. 2021;10(1):20.
4. Raut ABPD. NOSQL database and its comparison with RDBMS. International Journal of Computational Intelligence Research. 2017;13(7):1645-1651.
5. Han J, Haihong E, Le G, Du J. Survey on NoSQL database. In 2011 6th international conference on pervasive computing and applications. IEEE. 2011; 363-366.
6. Bjeladinovic S, Marjanovic Z, Babarogic S. A proposal of architecture for integration and uniform use of hybrid SQL/NoSQL database components. Journal of Systems and Software. 2020;168:110633.
7. Matallah H, Belalem G, Bouamrane K. Comparative study between the MySQL relational database and the MongoDB NoSQL database. International Journal of Software Science and Computational Intelligence (IJSSCI). 2021;13(3):38-63.
8. Ali W, Shafique MU, Majeed MA, Raza A. Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. Asian Journal of Research in Computer Science. 2019;4(2):1-10.
9. Chen JK, Lee WZ. An introduction of NoSQL databases based on their categories and application industries. Algorithms. 2019;12(5):106.
10. Gessert F, Wingerath W, Friedrich S, Ritter N. NoSQL database systems: a survey and decision guidance. Computer Science-Research and Development. 2017;32(3): 353-365.
11. Deka GC. A survey of cloud database systems. It Professional. IEEE. 2013; 16(2):50-57.
12. Okman L, Gal-Oz N, Gonen Y, Gudes E, Abramov J. Security issues in NoSQL databases. In 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications. IEEE. 2011;541-547.
13. Rao S, Suma SN, Sunitha M. Security solutions for big data analytics in healthcare. In 2015 Second International Conference on Advances in Computing and Communication Engineering. IEEE. 2015;510-514.
14. Hou B, Shi Y, Qian K, Tao L. Towards analyzing MongoDB NoSQL security and designing injection defense solution. In 2017 IEEE 3rd International Conference on Big Data Security on Cloud (bigdatasecurity), IEEE International Conference on High Performance and Smart Computing (hpsc), and IEEE International Conference on Intelligent Data and Security (ids). IEEE. 2017;90-95.
15. Abramova V, Bernardino J. NoSQL databases: MongoDB vs Cassandra. In Proceedings of the International C* Conference on Computer Science and Software Engineering. 2013;14-22.
16. Noiumkar P, Chomsiri T. A comparison the level of security on top 5 open source NoSQL databases. In The 9th International Conference on Information Technology and Applications (ICITA); 2014.
17. Dos Santos Ferreira G, Calil A, dos Santos Mello R. On providing DDL support for a relational layer over a document NoSQL database. In Proceedings of International Conference on Information Integration and Web- based Applications & Services. 2013;125-132.
18. Mason RT. NoSQL databases and data modeling techniques for a document-oriented NoSQL database. In Proceedings of Informing Science & IT Education Conference (InSITE). 2015;3(4):259-268.
19. Guimaraes V, Hondo F, Almeida R, Vera H, Holanda M, Araujo A, Lifschitz S. A study of genomic data provenance in NoSQL document-oriented database systems. In 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE. 2015;1525-1531.
20. Kumar J, Garg V. Security analysis of unstructured data in NoSQL MongoDB database. In 2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN). IEEE. 2017;300-305.
21. Tian X, Huang B, Wu M. A transparent middleware for encrypting data in MongoDB. In 2014 IEEE Workshop on Electronics, Computer and Applications. IEEE. 2014;906-909.
22. Zahid A, Masood R, Shibli MA. Security of sharded NoSQL databases: A comparative analysis. In 2014 Conference on

Information Assurance and Cyber Security (CIACS). IEEE. 2014;1-8.

23. Lim H, Fan B, Andersen DG, Kaminsky M. SILT: A memory-efficient, high-performance key- value store. In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles. 2011;1-13.
24. Zaki AK, Indiramma M. A novel Redis security extension for NoSQL database using authentication and encryption. In 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE. 2015;1-6.
25. Müller S, Bermbach D, Tai S, Pallas F. Benchmarking the performance impact of transport layer security in cloud database systems. In 2014 IEEE International Conference on Cloud Engineering. IEEE. 2014;27-36.
26. Sahafizadeh E, Nematbakhsh MA. A survey on security issues in big data and NoSQL. Advances in Computer Science: An International Journal. 2015;4(4):68-72.
27. Pallas F, Günther J, Bermbach D. Pick your choice in HBase: Security or performance. In 2016 IEEE International Conference on Big Data (Big Data). IEEE. 2016;548-554.
28. Vicknair C, Macias M, Zhao Z, Nan X, Chen Y, Wilkins D. A comparison of a graph database and a relational database: a data provenance perspective. In Proceedings of the 48th annual Southeast Regional Conference 2010;1-6.
29. Hurlburt G. High tech, high sec.: Security concerns in graph databases. IT Professional. IEEE. 2015;1:58-61.
30. Di Martino S, Fiadone L, Peron A, Riccabone A, Vitale VN. Industrial Internet of Things: Persistence for Time Series with NoSQL Databases. In 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE). IEEE. 2019;340-345.
31. Becker MY, Sewell P. Cassandra: Flexible trust management, applied to electronic health records. In Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004. IEEE. 2004;139-154.
32. Cuzzocrea A, Shahriar H. Data masking techniques for NoSQL database security: A systematic review. In 2017 IEEE International Conference on Big Data (Big Data). IEEE. 2017;4467-4473.
33. Lourenço JR, Cabral B, Carreiro P, Vieira M, Bernardino J. Choosing the right NoSQL database for the job: a quality attribute evaluation. Journal of Big Data. 2015;2(1):18.
34. Morgado C, Baioco GB, Basso T, Moraes R. A security model for access control in graph-oriented databases. In 2018 IEEE International Conference on Software Quality, Reliability and Security (QRS). IEEE. 2018;135-142.

APPENDIX

Table 1.

Name	Type	Security Issues							
		SQL injection	**XSS**	**Man in the middle attack**	**DOS**	**Undocumented service access**	**HTTP request smuggling**	**XXE**	**Code execution**
MongoDB	Document	**Yes**			**Yes**				**Yes**
Cassandra	Key-value	**Yes**		**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
CouchDB	Document	**Yes**	**Yes**		**Yes**				**Yes (RCE)**
Hypertable	Column								
Redis	Key-value				**Yes**				**Yes**
Riak	Key-value								
Neo4j	Graph	**Yes**						**Yes**	**Yes**
Hadoop HBASE	Column		**Yes**		**Yes**				
Couchbase	Document	**Yes**	**Yes**		**Yes**				**Yes (RCE)**
MemcacheDB	Key-value				**Yes**				**Yes**
RavenDB	Document	**Yes**							
Voldemort	Key-value								
Perst	Object- Oriented								
HyperGraphD B	Graph								
NeoDatis	Object- Oriented								
MyOODB	Object- Oriented								
OrientDB	Multi-model	**Yes**							**Yes**
Apache Drill	Supports all types of DB	**Yes**	**Yes**						
Amazon Neptune	Graph								

Table 2.

Name	Type	Security Issues							
		Bypass Somethin g	**Gain Privileges**	**Gain Informatio n**	**Directory traversal**	**Overflo w**	**Memory corruptio n**	**Cross-site request forgery (CSRF)**	**SSL**
MongoDB	Document	**Yes**		**Yes**					
Cassandra	Key-value								
CouchDB	Document	**Yes**	**Yes**	**Yes**	**Yes**				
Hypertable	Column								
Redis	Key-value	**Yes**		**Yes**		**Yes**	**Yes**		
Riak	Key-value								
Neo4j	Graph							**Yes**	
Hadoop HBASE	Column			**Yes**		**Yes**			
Couchbase	Document		**Yes**	**Yes**	**Yes**			**Yes**	**Yes**
MemcacheDB	Key-value	**Yes**				**Yes**			
RavenDB	Document								
Voldemort	Key-value								
Perst	Object- Oriented								
HyperGraphD B	Graph								
NeoDatis	Object- Oriented								
MyOODB	Object- Oriented								
OrientDB	Multi-model			**Yes**				**Yes**	
Apache Drill	Supports all types of DB			**Yes**					
Amazon Neptune	Graph								

Table 3.

Name	Type	Company names
MongoDB	Document	Uber, Lyft,LaunchDarkly, Delivery Hero, Stack, Accenture, Bepro Company, ViaVarejo
Cassandra	Key-value	Uber, Facebook, Netflix, Instagram, Spotify, Instacart, reddit, Accenture
CouchDB	Document	Awin, Our Stack, Digittal services
Hypertable	Column	Rediff.com, Ayima, baidu.com, Belvedere trading, Dehems, EPICS, Geliyoo, UCSF diagnoostics
Redis	Key-value	Atlassian, gardenbed, ASP.NET Boilerplate, Favorites
Riak	Key-value	Sentry, SendGrid, Nitanix, XING, Ambush Consulting
Neo4j	Graph	Unitedhealth Group, Nokia, Cisco, Northrop Grumman, National Geospatial-Intelligence Agency, KeyBank
Hadoop HBASE	Column	Pinterest, Hepsinbutada, Hubspot, JVM Stack, Awin, Tumblr
Couchbase	Document	Oxylabs, Agoda, UNIQLO, Immowelt AG, Checkout.com
MemcacheDB	Key-value	Facebook, Pinterest, Instagram, Twitter, Udemy, Shopify, Instacart, Slack
RavenDB	Document	Ezy, My Stack
Voldemort	Key-value	Apple, Peraton, Akamai Technologies, LiveIntent, Cyber Defense Solutions
Perst	Object-Oriented	Java applications, Microsoft .NET framework applications
HyperGraphDB	Graph	Server side and desktop java applications, Bioinformatics, Semantic web, Network research
NeoDatis	Object-Oriented	JConcept, NovaDutra, Tabula frame, Kasper Hansen, eSeller, Redmine android application
MyOODB	Object-Oriented	Developers, Financial and Insurance Industry, Information Technology, Science/Research
OrientDB	Multi-model	NVIDIA, Battelle, Mouser Electronics, Roblox, NV Energy, Idaho National Laboratory
Apache Drill	Supports all types of DB	JPMorgan Chase, Verisk Analytics, Unitedhealth Group, TeleTracking Technologies, HIS Markit, Deep Lens, Compile Inc, Clarisights, Alpha Vertex
Amazon Neptune	Graph	Thomson Reuters, NBC Universal, Herren, Geniusee, FetchyFox, extractBot, juncture,Industrial Inference, SEQL Tech Stack.

Peer-review history:
The peer review history for this paper can be accessed here:
https://www.sdiarticle5.com/review-history/87476