ISSN (Online) : 2454 -7190 **ISSN (Print) 0973-8975**

Copyright © J.Mech.Cont.& Math. Sci., Vol.-14, No.-1, January-February (2019) pp 402-413

Vulnerability Assesment For Advanced Injection Attacks Against Mongodb

[1]Vrinda Sachdeva, [2]Sachin Gupta

[1]Department of Computer Science Engineering, MVN University, Palwal, India

[2]Department of Computer Science Engineering, MVN University, Palwal, India

[1]vrinda08@gmail.com, [2]sachin.gupta@mvn.edu.in

Corresponding Author: Vrinda Sachdeva

Email: vrinda08@gmail.com

https://doi.org/10.26782/jmcms.2019.02.00028

Abstract

Nosql database is also known as not only sql database. For real time web application and for large set of distributed data, nosql database gaining popularity to handle big data. Nosql database has enormous function to handle big data. In contrast to this, nosql database also supports auto sharding, auto replication and many other feature making it suitable to be used as storage mechanism.

Nosql database is used to store data in an unstructured way, when more attention is paid to Performance and real time access rather than consistency, then nosql databases seems to be more appropriate. However, research on the security of nosql database is very limited. Although, there are many research benefit in nosql database like scalability, faster data access and availability as compare to rdbms. But nosql database has some security issues. The experimental testing on advance nosql injections is performed. The demonstration of advance nosql injection attack against a nosql database is performed with php and JavaScript. It shows the client's data. Method are discussed to prevent this type of security problems from happening again. This paper also shows how to create a security layer of nosql application to prevent nosql injection. In this paper, we will demonstrate, advance nosql injection attack and propose defense method to secure the nosql database. Hence nosql database programmer be aware of the nosql injection attack mechanism and build a more secure database to store huge data.

Keywords : Nosql,MongoDB,Injection,Attack,Consistency,Vulnerability,Scalability

I. Introduction

Nosql database are very important to store the huge amount of data for IT application. Nosql database came in existence after 2005 and it is better option for those problem that occur frequently in relational database. Nosql database is popular because it supports quick and fast data access. Many small and big companies are using nosql database because they are transferring their data into the cloud. In terms of database security, attack happens in both relational as well as non relational database. Auto sharing feature of nosql database provide load balance to perform the reliability and scalability during execution. Nosql database has four different data storage model. There are more than 225 nosql database available in the market for eg. MongoDB, Cassandra, redis. Many companies are using mongo DB and Cassandra.

II. Related Work

Many researchers have done lot of work in the area of nosql database security. Here we have reviewed and mentioned following references.

Okman, Lior [XV] presented two most popular nosql databases Cassandra and mongo db .It also outlines their main security problems and features. Architecture of mongo db and Cassandra is explained.

Ron, Aviv, Alexandra Shulman-Peleg, and Emanuel Bronshtein [III] presented an analysis of nosql threats. It also provide mitigation mechanism to minimize the attack. The main mechanism of sql attack is also relevant in nosql attack. It is divided into tautologies, union queries, piggybacked queries and java script injections.

EbrahimSahafizadeh, Mohammad Ali Nematbakhsh [VIII] described some privacy and security issue in big data and nosql database. Due to 3 V's i.e. variety, volume and velocity, there are lot of security and privacy issues. So security model of traditional relational database cannot deal with large scale of database. Therefore this paper presents some security and privacy challenges in big data. This paper presented an overview on big data and nosql database and described some security challenges in nosql database. Some nosql database have vulnerability for injection attack .So, needed to use sufficient input validation to avoid injection attack.

Varsha R Mouli, KP Jevitha [XX] present a survey on various web service attacks. It gives a systematic review on the studies of web service attack and security. There are many proposed solution to minimize the attack but no single solution exist to mitigate all the web service attack. Denial of service is the most common attack found on web services. This paper has analyzed 36 papers on web service attack. After survey, it is found that denial of service is the most common attack followed but XML injection attack. Therefore, Penetration and automation testing should be done on every development.

BoyuHou,Kai Qian [IV] explained many advantage of nosql database over relational database. It examined the security measures of mongo DB in terms of attack and defense. Experimental testing is also performed with the help of php and java script on nosql injection. From this, programmer will learn how to build a secure layer for

nosql application to prevent nosql injection. However security layer are recommended to be built to keep away the attack.

S.Priyadharshini,R. Rajmohan[XVIII] proposed a testing on nosql injection with the help of java script and php. It present a database protection system between dynamic application and database. Kerberos are designed for critical security in nosql database. It can be extended to give auditing services, thus securing to nosql db.

BoyuHou, yongshi[V] demonstrate server side JavaScript and HTTP injection attack and proposed defense method for the security of mongo DB. It will helps to nosql programmers to be aware of injection attack. Many malicious attacks are discussed in this paper.

III. Nosql Vulnerability

Every new technology has lacked security. Same is happened with nosql database .It has also lacked security when they first came in existence. Nosql databases suffered from problems like lack of encryption, role management, authentication, auditing,authorization. It allowed many dangerous attack like CSRF, denial of service .But, day by day situation is becoming better. New protection system has been developed. In nosql database sql injection attack are not possible because it use different query language. But still injection attack are possible in nosql database .Attacker can inject malicious query in nosql database. "The OWASP Top 10" mentioned some injection attack as security risk to the web application and also published recommendations to check the nosql injection code.

According to the OWASP Top 10 different type of injection attack are given below.

- Blind sql injection
- X Path Injection
- Blind X Path injection
- SQL injection
- SSI Injection
- OS command injection
- PHP injection
- LDAP injection
- I Frame Injection
- HTML Injection
- Format string attack
- Buffer overflow
- XSS
- XXE
- NOsql injection

For the current paper, we will focus on only Nosql injection Attack.

IV. Attack Mechanism Of Nosql Injection

Nosql injection refers to an injection attack by which hacker can enter malicious code

into input box in terms of nosql query. Thus Hackers get the information of the database by executing injection successfully. In nosql injection, attacker takes the benefit of unsanitized input character in nosql statement and inject arbitrary data into the query that will be executing the database engine as shown in fig. 1[II]

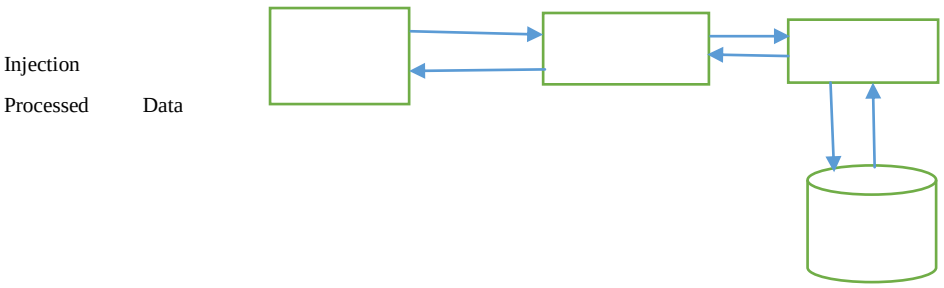

Fig. 1. Mechanism of Nosql attack

A. MONGODB NOSQL DATABASE

Mongo db is a document oriented database. It provides high scalability, good performance and intended to be scalable .It supports j son and b son data structure to store complex data type. It has powerful query language. It is used to store unstructured and semi structured data. It has very high speed to access the mass data. Although, It can handle million or billions of records. It is a real time database. Speed of mongo DB is 10 times faster than my sql. Because of this characteristics, many projects that are using large data are opting mongo DB instead of relational database. Many companies like eBay, Foursquare, Linked IN and others have adopted Mongo DB.

In mongo DB, queries and data are represented in JSON format which is more secure than sql. It is very simple to encode and decode the data in terms of JSON .It has the ability to run java script in the database engine to perform complicated queries like map reduce. Mongo DB nosql database, lacked security functionality when they first emerged [VI,III]. Mongo db is used by those projects that deals with big data. Day by day, Mongo DB is gaining more and more popularity for companies because it can store huge amount of big data. Injection occurs in input boxes as we mentioned in the previous section.

B. ADVANCE NOSQL INJECTION ON MONGODB

Mongo DB has been opted by more and more organisation for data management because of its speed and storage. In this paper, we will use injection in mongo DB. We will also analyse and detect injection in nosql database. Injection is a method by which hacker can attack on the database or even to crash the database. The detection of nosql injection can be understood through an example demonstrating the various nosql injection attack classification. Injection occurs on database by inputting a malicious code into the input box. By using this malicious code hackers gets user authority and obtain information from the database. Hackers inject malicious code

into input box .In the case of searching, the malicious code becomes a variable that participate in execution. Program starts to search on the basis of user input .When malicious code is executed then it always bring true results. As a result, hackers will get all data without passing input. Classification of nosql injection is given below.

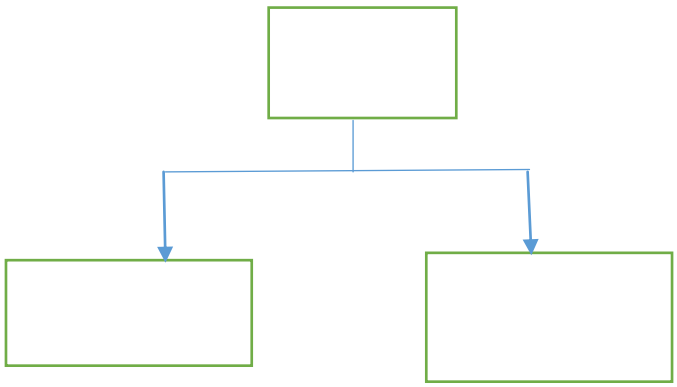

Fig.2. Classification of nosql injection

In this paper we will focus on advance nosql injection. Assuming, there is a student collection in mongo db which stores all the student data such as name, college _name, email _id, phone _no, location etc. Another collection is employee which stores employee id, name branch.
We mainly use injection by input box or by url using get or post method .For this, Assume that system is written by php and java script for passing the values and query to database. We can find the key without knowing their collection name by using advance nosql injection.

Injection no 1:

```
a'});return{something:1}}) //
```

Fig.3. advance nosql injection to get the key without knowing collection name.

After executing this Injection, the system gets the key.
We can also manipulate the key value without knowing exact value by using advance nosql injection.

Injection no 2:

```
a'}); return {name: 1}}) //
```

Fig.4. advance nosql injection to manipulate the key value

But we can only execute this injection after executing the injection no 1.Executing injection no 1, we get the name of all keys without getting their collection name. In

this injection we can manipulate the value of any key. In the given injection, we have manipulated the value of name key and it is set to 1.Now the next challenge is to get the database name. Although, it is very critical task to know the name of database.
By inserting this malicious code, we can also retrieve the database name. The results of find () function will always be true so that it will display the database name.

Injection no 3:

```
a'}); return {name:tojson(db.getName())}})//
```

Fig.5. advance nosql injection to retrieve the db name

After successfully executing this injection, we get the database name.
Now the next challenge is to get the collection name. Although, it is also very critical task to know the name of all the collection in the database.
But we can retrieve all the collection name that are stored in the database, directly on our system. For getting collection name injection is given below.

Injection no 4:

```
a'});return{name:tojson(db.getCollectionNames())}) //
```

Fig.6. advance nosql injection to get all the collections name

By getting all the collection name, Hacker can easily find the data of any collection. Basically, to execute injection, the main idea behind this is to make the condition statement always true so that the result can get passed when user search document wise and it will display all the information of the database to the user.

Injection no 5:

```
a'});return{name:tojson(db.employees.find()[1])}) //
```

Fig.7. advance nosql injection to find the data of any collection.

V RESULTS AND DISCUSSION

ALGORITHM

STEP 1: Get the key without knowing collection name.
STEP 2: Manipulate the key value without knowing exact value of key.
STEP 3: Retrieve the database name.
STEP 4: Retrieve all the collection name stored in the database**.**
STEP 5: Find the data of any collection in database.

Copyright © J.Mech.Cont.& Math. Sci., Vol.-14, No.-1, January-February (2019) pp 402-413

VI. MONGODB NOSQL DEFENSE AND DETECTION ANALYSIS

Injection could cause to spoof identity, repudiation issues like voiding transactions or manipulating balances .The hacker can do anything with data. Nosql injection attacks allows attackers to destroy the data or to make the data unavailable .Even he can become the administrator of the database. So, it is mandatory to add the function of defense to prevent from injection. In this way, the security of system will be improved.

For the defense approach, we have to avoid injection in the application from the input box or the url. After avoiding injection we will provide a detection approach should be based on the vulnerability of the threat level.

A. MONGODB NOSQL DEFENSE ANALYSIS

In this paper, for MONGODB defence we have mentioned 3 defence methods. The first defence method is input validation. It is used to limit the input entered by the user. The first one input validation which is to limit user input. For example, for the student database, when developer design the system, they can add code to limit the input so that it will accept only numbers. For phone number only numbers are allowed. So, we can add php code or java script code to satisfy this constraint .It will allow only number in phone number. The code is shown in fig.8

```
$id =$_GET [‘id’];
  If ( !preg_match (‘ / ^ [0-9] * $ / ’, $id ) )
        {
echo “ invalid number ”
        }
else
        {
echo “this number can be entered in phone number”
        }
```

Fig.8 Code to limit input entered by user

Every malicious code contains some notations. So, this method is used to avoid all unwanted notation. This method is used to sanitize the input.

The second defence method is to assign the permission to all user. By adding permission it can avoid the JS file injection. At the time of emergence of nosql database, it did not support authorisation, authentication and role management [XI]. But latest version of mongo application supports role management. While developing any application, developer need to ensure that the application should meet the basic security requirement. Firstly, application should be able to identify the user.

Secondly, only authorize user must be able to access the database.

In the department of college, the role are divided into student, teacher and parent. Every user get appropriate permission depends on their role. It helps in authorization process .For example in the attendance management system student can check their attendance information, but they are no allowed to update the attendance. They can only view their attendance record.

On the other hand, teacher can update the attendance of student. Teacher also maintain the record of detained student .So, in the role based system, authorization is directly linked to their role. There are different permission level according to their roles.

The third defence solution is the parameterized statement .The solution is to check and filter the variable of a parameterized statement. For user input parameterized are used to pass variable .This method prohibits to embed user input variable in the query. Using this method, we can prevent from most of harmful injection attack. User also pass $_POST parameter to a query. $_POST is more secure as compare to $_GET.

With the help of some code, we can easily determine the character of variable .It helps to check whether it contains number only or not. If yes then it is secure code otherwise we can reject.

```
include("includes/conn_mongo.php");

$search = $_POST['search'];

$collection=$db->student;
$query="var data = db.student.findOne({phone:'$search'});return data;";
$cursor=$db->execute($query);
?>
<?php
if(is_numeric($cursor)==’true’))
{
        if($cursor['retval']!=NULL)
        { ?>
<tr>
        <td><?php echo $cursor['retval']['name']; ?></td>
<td><?php echo $cursor['retval']['email']; ?></td>
<td><?php echo $cursor['retval']['phone']; ></td>
<td><?php echo $cursor['retval']['college']; ?></td>
```

```
<td><?php echo $cursor['retval']['country']; ?></td>
</tr>
<?php   }
else
{
echo "incorrect";
}
}
```

Fig.9 Call for parameterized statement

Based on the above discussion, it is necessary to create a system that check input validation, assigning permission, permission level and parameterized statement. If all these security requirements are included in the code, then the system will be much secure.

B. MALICIOUS FEATURE DETECTION

From the last few years, security experts are paying more attention to attacks. Every web application developer have to follow the security code while developing the application. After paying so much effort to security code, attacker can still hack the system. If there is a single point of failure then all effort made by developer for secure the code will be useless.
Malicious feature detection approach find if the system has feature that are harmful for security .Malicious feature are given below.

1. Special meta character
2. Malicious command
3. Command modifier

This malicious feature detection chart help developers to detect malicious code in the statement. If these malicious feature not detected, then web application blindly pass these malicious feature to the system. This malicious feature detection helps to make secure application.

VII. Conclusion and Future Work

Information security play most important role in system. Most of the databases are vulnerable. There are always some vulnerability that are used for malicious purpose. Using those vulnerability hacker can attack on the system. Some injection attack are discussed in this paper. Algorithm is mentioned to find the injection attack. This safety algorithm helps to enhance appropriate security mechanism for making more secure application. Hackers can easily leak the information. So, to achieve the confidential information, data must be integrated with various security measures through technical means.

In this paper, we have discussed advance nosql injection. In future work we will analyses some more advanced nosql injection possibilities on MONGODB, as well as to study how to defense and mitigate attack to make the system more secure.

References

I. Abramova, Veronika, and Jorge Bernardino "No SQL databases: Mongo DB vs Cassandra." Proceedings of the International C* Conference on Computer Science and Software Engineering 10 Jul: 14-22, 2013.

II. Ahmed M. Eassa , Hazem M. El-Bakry "No SQL Racket: A Testing Tool for Detecting No SQL Injection Attacks in Web Applications" International Journal of Advanced Computer Science and Applications, Vol. 8, No. 11, 2017

III. Aviv Ron, Alexandra Shulman-Peleg, Emanuel Bronshtein "No SQL, No Injection? Examining No SQL Security Examining No SQL Security" In proceedings of the 9th workshop on web 2.0 security and privacy (W2SP) 2015

IV. BoyuHou,Kai Qian "Mongo DB No SQL injection Analysis and detection" 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing

V. BoyuHou,yongshi" Towards analyzing Mongo DB No SQL security and designing injection defense solution" ieee 3rd international conference on big data security on cloud (big data security), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids), 26-28 may 2017.

VI. Chickerur, Satyadhyan, Anoop Goudar, and AnkitaKinnerkar "Comparison of Relational Database with Document-Oriented Database (Mongo DB) for Big Data Applications." 28th International Conference on Advanced Software Engineering & Its Applications (ASEA) 25 Nov. 2015: 41-47.

VII. Changlin He,“ Survey on nosql database technology”, journal of applied science and engineering innovation vol. 2 no. 2,2015

VIII. EbrahimSahafizadeh, Mohammad Ali Nematbakhsh **“** A Survey on Security Issues in Big Data and No SQL” ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 4, No.16 , July 2015 ISSN : 2322-5157

IX. https://www.mongodb.org

X. Jing Han,haihongE,GuanLe,JianDu,“ survey on nosql database”, 2011 IEEE.

XI. Kadebu, Prudence, and Innocent Mapanga, "A Security Requirements Perspective towards a Secured NOSQL Database Environment." International Conference of Advance Research and Innovation, 2014.

XII. ManovegSaxena, ZakirAli, Vinod Kumar Singh,“ NO SQL database – analysis, Techniques and classification” journal of advanced database management &system, volume 1 issue 2,2014.

XIII. Noiumkar, Preecha, and TawatchaiChomsiri,"A Comparison the Level of Security on Top 5 Open Source No SQL Databases." The 9th International Conference on Information Technology and Applications (ICITA2014).

XIV. “No SQL Injection in Mongo DB” https://zanon.io/posts/nosql-injection-in-mongodb.

XV. Okman, Lior et al, "Security issues in nosql databases." 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications 16 Nov. 2011: 541-547.

XVI. Pokorny, Jaroslav, "No SQL databases: a step to database scalability in web environment." International Journal of Web Information Systems 9.1 :69-82,2013.

XVII. Roshni Bajpayee,Sonalipriya Sinha,Vinod Kumar ,“Big data :A brief investigation on NOSQL database”, International journal of innovations & advancement in computer science, volume 4, issue 1 January 2015.

XVIII. S. Priyadharshini, R. Rajmohan “Analysis on data base security model against nosql injection” 2017 International Journal of Scientific Research in Computer Science, Engineering and Information Technology , Volume 2 , Issue 2 ,2017, ISSN : 2456-3307

XIX. Sharma, Chandershekhar, and SC Jain, "Analysis and classification of SQL injection vulnerabilities and attacks on web applications." Advances in Engineering and Technology Research (ICAETR), International Conference on 1 Aug. 2014.

XX. Varsha R Mouli, KP Jevitha "Web Services Attacks and Security- A Systematic Literature Review " 6th International Conference On Advances In Computing & Communications, ICACC 2016, 6-8 September 2016, Cochin, India

XXI. VatikaSharma,Meenudave," SQL and NOSQL databases", International journal of advanced research in Computer science and software engineering, volume 2 issue 8,August,2012 .