IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol.-7, Special Issue-9, April 2019 E-ISSN: 2347-2693

A Novel Approach on Tech Solutions to Mitigate Big Data Security Threats

Suneetha V[1], Sunitha. M[2], Arshiya[3]

[1]Rayalaseema University, Kurnool, Andhra Pradesh, India
[2,3]Dayananda Sagar College of Arts Science and Commerce, Bangalore, India

[*]*Corresponding Author: hod-mcabu@dayanandasagar.edu, Tel.: +91-9632831002*

DOI: https://doi.org/10.26438/ijcse/v7si9.6368 | Available online at: www.ijcseonline.org

Abstract— Big Data is the buzz word today. Numerous projects centering big data are booming. It hides in itself vital information which if unearthed would provide great insights into various areas. The growing need of this data is complicating security. Handling big data projects is a challenging task. To prevent a potentially disastrous data breach, big data security should be considered seriously. To prevent security issues, some simple steps should be implemented. It can be done with best practices and internal controls, like protecting against *NoSQL Injection* points. These points provide a way for attackers to access Big Data. This paper discusses the various big data security threats and different innovative Tech solutions to meet many of the security concerns hindering the Big Data persistence, analysis and presentation.

Keywords— *Big Data, Big data security threats, Data Security, SQL injection, NoSQL injection.*

I. INTRODUCTION

The Big data is mainly evolved from any kind of source that consists of huge volumes, velocity and various variety of data. It helps to gather, collect and store the information, manages and analyze, with vast amount of data with proper speed and right time. In terms of complex, and noisy data big data helps to handle in an efficient way. To handle such huge amount of data it has become a great challenge. The big data analytics helps the companies to analyze many types of data such as structured , unstructured in terms of streaming data and also the combination of both that is semi-structured data. Big data has made valuable changes in the industry in areas like health care, finance, banking fraud detection, credit management etc, to provide faster, proper valuable services due to which the government sectors started focusing. IT owns the raw data and business units started taking the responsibility to provide valuable ownership. To classify the information is even more critical. For big data owners always security breach will become major issue.

The paper is described as follows. Importance of bid data are discussed in section I, The important challenges are given in Section II, In section III security issues are discussed, followed with existing methods and tech solutions of big data in section IV. Finally the paper is concluded with some solutions in section V.

II. CHALLENGES IN BIG DATA SECURITY

The various important challenges related to Big data security are discussed below:

A) Big data security challenges related to characteristics:

Fig-1: Big Data security challenges

Data Volume: The Volume of Big data is increasing every nanosecond. The main source which is generating a large volume of data is social media. The data is generated from Petabytes to Zeta bytes. It is very hard to maintain such large data.

Data Variety: Since the data generated in big data includes structured, semi-structured and unstructured data (which can be in any format i.e. audio, video, text images, etc.), it increases data complexity.

Data Velocity: There is a large amount of data that is coming in and out of a system with a very high speed. There is no exact technology that can deal with this data overflow.

Regulatory requirements: It is very important that the data which is stored as big data is problem specific and goal oriented. As supervised learning requires accurate data to infer correct results, failing to which it may lead to inaccurate results.

Data Veracity: Random data, Uncertain input data and approximate modelling lead to data veracity challenge.

Application Specific Security: Many users access the same application without having proper authentication to it, causing security threat to the application.

Framework Specific security: Providing the required framework for all the data is not an easy task. As each and every task requires an independent framework

Security for data during rest, processing, presentation: Big data includes private data which has to be secured during rest, processing and presentation which is again a difficult task because of its large volume.

B)Technical Challenges

Big data in today's world combines the rapid increase usage various types of data from different sources which is characterized into many forms. Due to this privacy and security are the major problems to deal with. RDBMS will not support to handle such streaming data because it is structured. NOSQL databases gives a proper method to deal with streaming data. It also helps for storage and retrieval purpose. The NOSQL databases are characterized into four types. 1. Key-valued, 2. Document Oriented, 3.Graph oriented and 4. Column oriented databases. The companies depend upon their input data they use specific NOSQL database to provide insights for their companies. To handle such streaming data NOSQL data bases provide main feature called Scalability to provide great performance. Some of the challenges in terms of security are Integrity, availability, confidentiality are the important elements of security. Encryption is one of the important aspect of security. The NOSQL databases deals with BASE properties .The access controls and techniques such as attribute based encryption are important to protect the sensitive data. Big data must accept and support multiple layers of security in terms of the data which is at rest and in motion.

III. SECURITY ISSUES IN BIG DATA

Some unique security issues in Hadoop are encountered below:

1) Split Data:

The clusters of Big Data contain the data that represent the quality by allowing multiple copies transferring from one node to another which ensures redundancy and resiliency. The split data is available to share across multiple servers. Due to the absence of security model fragmentation leads to security issue.

2) *Distributed Computing*:

The data is distributed over the cluster of nodes. At any instant the data is available. To deal with huge amount of parallel computation the environment is complicated created with risks of attacks. These risks enable more security issues.

3) *Data Admittance*:

Majority of the companies generate sensitive information. There is a risk of unauthorized access to physical and logical systems. Access control is the important component that ensures security. At the first stage itself the data is addressed in terms of access related scenarios.

4) Communication in the large cluster:

As the data is distributed in large cluster of nodes a major concern of Hadoop occurs. They don't implement proper secure communication between node - to- node. This leads a major security issue.

5) Communication and Interaction with Client:

The client communicates with the resource manager and data nodes. There should be an efficient communication between Client and data nodes. Managing the protected nodes from clients and Name servers is difficult. There is a chance to propagate harmful data in terms of client.

6) SQL Injection

Database security is a vital aspect of protecting the information. To access companies data attackers have control over the data. The SQL injection attacks adds harmful data into the database layer through various statements. The attackers can access, alter, delete and insert all the unofficial data. The SQL injection declines over the years due to good frame works. The SQL injection also allows the attackers to take off user's identity, alter with existing data cause

Some issues such as negating transactions allow the whole data on the system. It also destroy the data to make it otherwise unavailable. It becomes main administrators of the database server.

The script below is executed on a web server. To authenticate with a username and password this script is a good example. The example database has a table named users with the following columns:


```
# Define POST variables
uname = request.POST['uname']
pwd = request.POST['pword']
# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE uname='" + uname + "' AND pword='" + pwd + "'"
# Execute the SQL statement
database.execute(sql)
```

These input fields are vulnerable to SQL Injection. An attacker could use SQL commands in the input in a way that would alter the SQL statement executed by the database server. For example, they could use a trick involving a single quote and set the `pwd` field to:

```
pword' OR 1=1
```

As a result, the database server runs the following SQL query:

```
SELECT id FROM users WHERE uname='uname' AND pword='pword' OR 1=1'
```

*7. **NOSQL Injection:***

NOSQL databases handles the streaming data. To deal with the data of this kind security is the major issue. The databases are challenged in terms of encryption, proper support and some fine grained permission. Due to this there are chances of unsafe network and attacks. There are different NOSQL databases to handle such attacks. These databases use various query languages .These are immune to injections. Security challenges are improved . Many applications and services use NOSQL databases mainly to store the user's data. The access is provided to database via driver. A driver is a protocol wrapper which gives libraries to access multiple languages. Some times the drivers itself are not secure enough to handle such data. The attackers can boat a web access request with an injection.

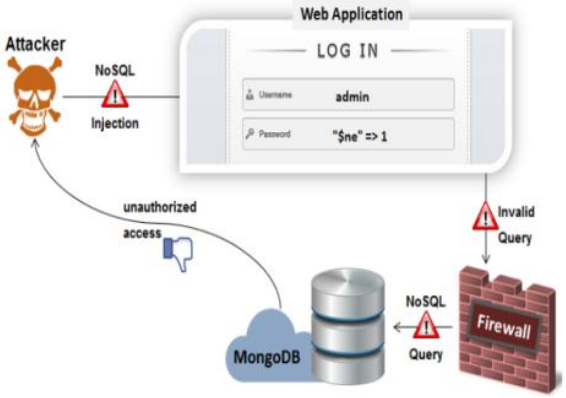

Fig-2: NoSQL Attack Vectors:

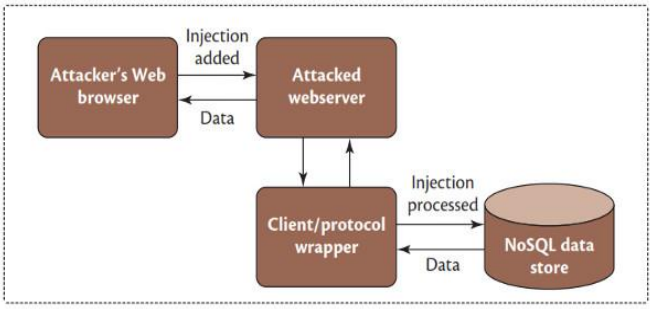

Fig-3. A Typical Web application architecture.

IV. PROPOSED SECURITY SOLUTIONS FOR BIG DATA

Huge data sets dealing with parallel computations and input data from various sources are supported by Hadoop framework . The framework supports all types of data in terms variety, velocity and volume which is aggregated as structured, unstructured and semi structured data. The enterprises should access security to data to extract maximum value in terms of analytics. To deal with such data there is high risk . The insightful data is exposed. To initiate big data and get the solution to give security of the sensitive data enables analytics for good insights. The following are some requirements for the solution:

a. To secure this data in big data systems and give access control to data protection by monitoring all the details.
b. To guard the data and to maintain for accurate analytics in its encrypted form.
c. To present mass layer encryption to progress security and to enable clusters of data to scale up accordingly.
d. To control security tools NOSQL clusters are built
e. According to business requirements the business is scaled up. The solutions are architecture to improve the growth of business.
f. The protection should be given to users from complexity of security.
g. The solutions in terms of data protection need to work independent of complex reengineering dealing with environments of IT.
h. Cloud technology is utilized to give protection of data.

4.1 *Security models:*

A) Data De-Identification Model

At initial level the data should be de-identified. Across the country many government agencies share the dataset of information to analyze the data for emerging risks. During the production itself data is protected. There should be a steady encrypt and decrypt operations. This model can counteract the data breaches by portraying the data which has no value. The important techniques used in data-identification model a tokenization, encryption and data masking.

B) Unique Approach to Security

This approach towards the security gives a method to deal with protection of the data. Where ever the data goes the protection is provided. The security is mainly focused on infrastructure elements mainly servers, networks and databases. This method handles the data at risk. In case of any breach the actual data is exposed in the event. The data centric security approach is mainly used to capture the data protection by an application making the data which is not used to attackers. This approach gives the protection at the stages of rest, motion etc. The data can be unmasked by proper users who are authorized based on their need. The access is highly controlled with proper management.

C) Walled Garden Model

This model specifies covering the security into the application.

The main purpose of this model is to place the group of systems into its own network. It is done by accessing through various techniques like firewalls and also controlled access. It gives proper security for the cluster by giving security. The demerit of this model is it does not provide the means to prevent the user credentials from misuse .The users cannot view the modified data which is stored in various systems.

D) Jujutsu Security

This security is originated from the martial art method. It represents manipulating the opposite person energy against himself instead of fighting with the enemy. The bigdata handling vast amounts of variety data relies on the capability to design and apply a specific engine which is dynamic that recommends the insightful data in the system. The Jujutsu security model helps to design Big data security in an efficient way.

4.2 Tech Solutions for the Big Data Safety & Security:

The major issue, tackled by big data analytics with respect to data protection sight is ensuring the data security and its confidentiality. Big Data analytics will provide the deep and helpful insights from the data, but its major concern in this process is protecting privacy of sensitive information against data breaches as it holds huge amount of private data. Data breaching may affect in much more critical way and lead to disturbing consequences than we expect and see usually. Thus, there is a crucial need for robust Tech solutions to protect data from all above security breaches.

Big data Security – Best Practices to avoid security issues

1. ***Vetting cloud service providers:***
   If your big data is stored in the cloud, you must assure that your provider has adequate shield mechanism in place. Ensure that the provider does periodic safety audits and agree on consequence in the situation when sufficient security principles are not met.
2. ***Safeguard your data:***
   It is very vital to guard both your data that is raw data and the upshot from analytics. To ensure no key data is seeped out, encryption should be used consequently.
3. ***Adequate access control policy:***
   Strategies should be made such that they allow access to endorsed users only. This will prevent unofficial access to data from both internal and external sources.
4. ***Network protection:***
   Sufficient security can be provided for the data getting transmitted via network to guarantee secrecy and reliability.
5. ***Real-time security control:***
   A supervised control on the data access is very much needed. Threat intelligence can be used to avoid unofficial data access.

Some of the Tech solutions are discussed below:

1. ***SQL Injection Mitigation****:*
   We can mitigate or avoid the SQL injection occurrences by using input data authentication and parameterized query statements. We should avoid the direct input in the application coding. The programmer should filter all inputs, along with web page input forms like login pages. We should remove vulnerable code elements like single quotes and etc. Fair idea is to hide visibility of DB errors. SQL Injection can make use of database bugs to gain details regarding your database.

a) SQL query statements with parameters:-- PreparedStatement object

Parameterized SQL statements can strongly mitigate the security attack. Runtime query statements fails to distinguish among program code and data. SQL statements with runtime parameters allow programmers to execute static SQL query by passing external parameters as input to the query. In this process, the SQL interpreter constantly distinguishes application code and data.

a*uthenticate()* method using a runtime parameters feature is as below:-Prepared Statement object

```
Public Boolean authenticate (String name, String pass)

{

PreparedStatement pstmt;

String sql = "SELECT name FROM user WHERE name = ? AND passwd =? ";

pstmt = this.conn.prepareStatement(sql);

pstmt.setString(0, name);

pstmt.setString(1, pass);

ResultSet results = pstmt.executeQuery();

return results.first();

}
```

Fig-4: Example of PreparedStatement

Irrespective of input from the user, dynamic parameters *name* and *pass* won't influence the actions of the

sql statement. Merely using *PreparedStatement* object only can't resist SQL injection attacks. It supposed to be patched all along with parameterization aspect ("?") for all dynamic parameters. Usage of *PreparedStatement* object alone cannot serve the purpose if we are not using parameterization.

b) Stored procedures:

These well defined and stored chunks of SQL statements are triggered by the application code. Programmers design and write SQL query statements with automatic dynamic parameters. It's feasible for a programmer to write dynamic SQL query statements within stored procedures.

c) Input validation:

A usual resource for SQL injection is malicious intended external input. Off course, at all times it is an excellent coding standard to only allow permitted data input via input validations like blacklist validation technique and whitelist validation method. Blacklist validation checks the user input data with a set of recognized suspectable or intended inputs. A program enlists all possible indented data inputs, and then verifies and validates the user data input aligned with the prepared list. But, an attacker can easily escape from the Blacklist validation techniques by applying an alternative malicious data input which not part of the programmer's prepared list.

Whitelisting could be a far better technique to mitigate the SQL injection risk. Whitelist technique checks user data input against a set of well-known, authorized input. In this validation, the program knows clearly what is correct and incorrect input values. So it rejects the malicious input.

d) Principle of least privilege:

This is a typical safety measure which assists to lessen the possible loss of a triumphant attack. Program shouldn't allow or give DBA or admin grants or permission upon the DB server. Furthermore, based on requirement necessities, privileges can be allotted. For example, One need read permission are only granted read access to the table. This ensures that if an application is compromised, an attacker won't have the rights to the database through the compromised application.

2. NOSQL Injection Mitigation:

Mitigating security risks in NoSQL deployments is significant in light of the attack vectors. Let's examine a few suggestions for each of the threats:

1. Prepared statements should be used instead of building dynamic queries using string concatenation. Strong JSON structure queries
2. ***Input Validation:*** Validate inputs to detect malicious values.
3. ***Principle of least privilege:*** To minimize the potential damage of a successful injection attack, do not assign DBA or admin type access rights to your application. Similarly reduce the privileges of the operating system account that the database process runs under.

4. Security scanning to prevent injections

In order to mitigate injection attacks it is suggested to use out of the box programming tools while building queries. For JSON queries such as in MongoDB and CouchDB almost all languages have good native encoding which will finish the injection risk. It is also advised to run Dynamic Application Security Testing (DAST) and static code analysis on the application in order to find any injection vulnerabilities upon not incorporating coding guidelines. The problem is that most of the tools in the market today still lack methods for detecting NoSQL injections. DAST methodology is considered more reliable than static analysis, particularly if used in combination with some backend inspection technology that improves detection reliability, a methodology referred to as Interactive Application Security Testing (IAST).

5. Access Control and Prevention of Privilege Escalation

Earlier NoSQL did not support proper validation and role management, but today it is possible to manage proper validation and RBAC authorization on most popular NoSQL databases. Utilizing these methods is significant for two reasons. First, they allow enforcing the principle of least privilege thus preventing privilege escalation attacks by genuine users. Second, similarly to SQL injection attacks, proper privilege isolation allows to lessen the damage in case of data store exposure via the above portrayed injections.

V.CONCLUSION

Big data industry is rapidly developed. The evolution of technology and the innovation in applications are advanced with the increasing speed. With this rapid development new forms of data storage, distributed and parallel computing developed. Big data security should be considered seriously and appropriate measures must be taken to prevent a disastrous data breach. The challenges in security should be handled with well-organized tools and policies which help to protect the applications. In this paper we have enlisted all the security models to ensure the data security. Providing robust security model in dynamic big data environment is crucial and long way to go but understanding the existing security models is helpful to design secure systems. We made effort to discuss the possible best practices and tech solutions to mitigate prominent security issues like Data Admittance, SQL Injection, NOSQL Injection and etc.

REFERENCES

[1] Ahmed E. Youssef and Manal Alageel, "A Framework for Secure Cloud Computing", International Journal of Computer Science Issues (IJCSI), Vol. 9, Issue 4, No 3, pp. 478-500, July 2012.

[2] A. Youssef and M. Alageel, "Security Issues in Cloud Computing", the GSTF International Journal on Computing , Vol.1 No. 3, 2011.

[3] P. Groves, B. Kayyali, D. Knott and S. V. Kuiken, "The 'big data' revolution in healthcare," McKinsey &Company, 2013.

[4] P. Institute, "Third Annual Benchmark Study on Patient Privacy and Data Security," Ponemon Institute LLC, 2012..

[5] Shoffner, M. (2012). Handling "hot" health data without getting burned. Presented at the Strata Rx Conference, San Francisco, CA, USA. http://strataconf.com/rx2012/public/schedule/detail/26231..

[6] Katal, A.; Wazid, M.; Goudar, R.H., "Big data: Issues, challenges, tools and Good practices," Contemporary Computing (IC3), 2013 Sixth International Conference on , vol., no., pp.404,409, 8-10 Aug. 2013.

[7] Nambiar, R.; Bhardwaj, R.; Sethi, A.; Vargheese, R., "A look at challenges and opportunities of Big Data analytics in healthcare," Big Data, 2013 IEEE International Conference on , vol., no., pp.17,22, 6-9 Oct. 2013.

[8] Sun, Jimeng, and Chandan K. Reddy. "Big data analytics for healthcare." Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2013.

[9] Smitha Rao, S.N.Suma, M.Sunitha. "Security Solutions for Big Data Analytics in Healthcare",2015 Second International Conference n Advances in Computing and Communication Engineering,2015.

[10] www.whishworks.com

[11] http://blog.sqlauthority.com

[12] www.ibmbigdatahub.com

[13] https://arxiv.org/ftp/arxiv/papers/1506/1506.04082.pdf

[14] "Analysis and Diminution of NoSQL Injection attacks" by Pritesh, Ashwini, Rachana Badekar

[15] https://thesai.org/Downloads/Volume8No11/Paper_78-NoSQL_Racket_A_Testing_Tool.pdf

[16] https://pdfs.semanticscholar.org/9e33/338f42d2d97349063cf24db36a74936f0613.pdf

[17] https://www.synopsys.com/software-integrity/resources/knowledge-database/sql-injection.html
https://www.acunetix.com/websitesecurity/sql-injection/

[17] www.synopsys.com

[18] www.acenutix.com

AUTHORS PROFILE

Mrs Suneetha V is working as HOD-MCA Department, Dayananda Sagar College of Arts, Science and Commerce. She has overall experince of 19 years. She is pursiung her research at Rayalaseema University, Kurnool. Her area of research is Privacy Preservation in Big Data.

Mrs. Sunitha M is working as Asst Professor in Department of MCA , Dayananda Sagar College of Arts, Science & Commerce. She has overall teaching experience of 10 years.Her area of research is Security and Privacy Solutions in Big Data and Cloud computing.

