Impact Factor (SJIF): 4.542

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 4, Issue 4, April-2017

Identifying Vulnerabilities in Apache Cassandra

[1] Sneha B. Chaudhari, [2]Ravi K. Sheth,
[1]Student, M.tech (cyber security), [2]Assistant professor (IT)
[1], [2,] Department of Information Technology
[1,2]Raksha Shakti University, Gujarat-Ahmadabad, India.

ABSTRACT

Database play important role in any system. Now a day's all the applications are global so data can be not located at single location. The database can be distributed and to handle such system, Distributed Database Management system is required. The distributed database management system must be very powerful system which can handle simultaneously many users effectively. Apache Cassandra is such powerful system but there are certain vulnerabilities in apache Cassandra. This paper focuses on identifying the vulnerabilities of Apache Cassandra. This paper also focuses on security policies in Apache Cassandra and describes how these vulnerabilities can be avoided using security policies effectively.

KEYWORDS: - Apache Cassandra, DDBMS, RDBMS

1. INTRODUCTION

Cassandra is an open source was originally designed by Facebook, Inc. to support user searches of their Inbox. Cassandra was able to support the millions of users using Facebook's Inbox search application at any one moment. Cassandra also enables the distribution and replication of data amongst a large number of servers so that data is highly available [1].

1.1 Distributed Database

Distributed database mean database in which portions of the database are stored in multiple physical locations and processing is distributed among multiple database nodes. In distributed database, the databases are homogeneous or heterogeneous. [2] To manage distributed database, effective distributed database management system (DDBMS) is required as there are many roles of DDBMS. DDBMS should provide fast data retrieval on single query of user and data should be consistent. The DDBMS plays major role in distributed database. Apache Cassandra is one of the important DDBMS.

1.2 Apache Cassandra

Apache Cassandra is a high performance, very scalable, fault tolerant, distributed post relational database solution. [3]It combines all benefits of Google Bigtable and Amazon Dynamo to handle the types of database management that traditional RDBMS vendors cannot support .[3]

Cassandra is using keyspace, which is collection of tables and tables are collection of partitioned rows. [4]The table of Cassandra is almost similar to the table of MYSQL, where table represents an entity and rows represent instances of that entity. Each row has a primary key and zero or more clustering keys.[4] Cassandra, however does not require a rigid schema [5].

1.3 Data modeling in Cassandra

Cassandra models entities as tables of partitioned rows with each row representing an instance of the entity [6]. Each row has either a simple primary key (the first column in the table) or a compound primary key (the first two or more columns in the table) [6]. The first column is also known as the partitioning key as it determines up on which node Cassandra will store the row's data [6]. The remaining columns(known as clustering columns) in a compound primary key determine how the rows are sorted on a node [6]. Cassandra's data model is based on denormalization [6].

1.4 Vulnerabilities in Apache Cassandra

Authentication and authorization are disabled by default. After enabling authentication the default user name and password are Cassandra/ Cassandra. The Cassandra user is super user. The (Database Administrator) DBA can create more Cassandra accounts.[4] The admin can grant privileges to users to limit the access that a use has within the database. The non-super users can only read some tables from 'system keyspace'.

There is no default encryption facility in communication. The encryption can be enabled through the configuration file which require SSL encryption for both. Cassandra does not support built in facility for encryption.

Cassandra supports OpsCentre that is web based tool for accessing diagnostic and administrative information related with it. By default there is no authentication but this can be enabled explicitly.

2. DATABASE VULNERABILITY OVERVIEW AND METHODOLOGY

There are several database vulnerabilities like injection, misconfigured databases, HTTP interface, encryption, authentication and authorization.

.

2.1 Injection

Injection occurs when a user provides input for a SQL or NoSQL query that changes the intended meaning of the query. The root cause of injection attacks is a failure to properly validate user input.
There are various categories of injections like tautologies, illegal or logically incorrect queries, union queries, and piggy-back queries.
This paper focuses on its vulnerabilities in apache Cassandra.

2.1.1 Tautologies

Cassandra is not vulnerable to this category of injection as the Python and Java drivers for Cassandra detect a syntax error when attempting to parse a comment marker ("**--**" for CQL).

2.1.2 Illegal or Logically Incorrect Queries

Cassandra does not allow sub-queries, therefore it also does not yield the level of vulnerability that MySQL has regarding illegal and illogical queries. However, both the Python and Java presentation tiers yield useful information in their syntax error messages.

2.1.3 Union Queries

Cassandra is not vulnerable to this category of injection because the query language does not support union queries.

2.1.4 Piggy-back Queries

Cassandra is not vulnerable to this category of injection as the Python and Java drivers for Cassandra detect a syntax error when attempting to parse a comment marker ("**--**" for CQL).

2.2 Misconfigured Databases

Security misconfiguration is also a critical risk. This vulnerability refers to the failure to change the default settings of a database management system like default accounts, network connection settings, and security settings.

The default networking configuration is not vulnerability for Cassandra. Cassandra does use a default port for CQL clients (9042), but the default bind address is localhost [4]. Cassandra however, have a configuration vulnerability related to server encryption. When using encryption, the Java Keystore is used to store the private key that Cassandra utilizes for encrypting outgoing messages. The Truststore is used to store the trusted certificate that Cassandra utilizes to authenticate remove servers. The default password for both the Java Keystore and the Truststore is “Cassandra.” [4] The mitigation for this vulnerability is changing the default password, which can be specified in the Cassandra configuration file.

2.3 HTTP Interface

This vulnerability refers to any interface that the database management system offers that allows someone to collect any data concerning the database. This could include administrative information, schema definitions, and the actual stored data.

Datastax Community Edition provides OpsCenter as an installation option along with Cassandra. OpsCenter is a web based tool for accessing diagnostic and administrative information relating to the Cassandra database [7]. The configuration file for OpsCenter is separate from the configuration file for Cassandra. By default the OpsCenter bind address is all network interfaces (0.0.0.0) on port 8888 [7]. This differs from Cassandra’s default bind address, which is 127.0.0.1 (localhost) [6]. By default there is no authentication or encryption for OpsCenter, but this can be enabled in the configuration file [7].
OpsCenter provides both information and utilities for the database administrator to interact with the cluster and keyspaces within the cluster. An individual accessing OpsCenter can see the version of Cassandra, a list of all the nodes in the cluster along with their IP addresses, a list of the keyspaces stored in the cluster, a list of the tables stored in each keyspace, and the CQL used to define each table. The user can take several actions through OpsCenter directed at the cluster or individual nodes. The user can change any of the configuration settings for the cluster to include the port, the bind address, authentication options, authorization options, and encryption options. The individual using OpsCenter can also delete the cluster. OpsCenter also enables the user to configure a node, stop a node, restart a node, decommission a node, and even add nodes. If authentication is enabled for Cassandra, but not OpsCenter, a malicious user can change Cassandra settings to remove the requirement for authentication.

2.4 Encryption

Sensitive data exposure or loss of data can result from lack of encryption. This vulnerability includes both the encryption of data transmitting between two entities like client /server and the encryption of data that is stored in the database [8].

Cassandra does not enable encryption by default for client to server communication, node to node communication, or client to node communication [4]. In order to mitigate this, the database administrator can modify the configuration file to enable TLS/SSL encryption for client to server, node to node, and client to code communications. The HTTP interface for Cassandra, OpsCenter, does not enable encryption by default; however the database administrator can enable encryption through the OpsCenter menu [7]. Built-in functions do not exist in Cassandra to encrypt data for storage in the database [1].

2.5 Authentication and Authorization

Lack of Authentication and authorization for specific functions is a serious risk. These two vulnerabilities are related, but distinct from one another with authentication concerning itself with verifying a user's identity and authorization concerning itself with verifying a user's privileges or granting privileges to a user

Cassandra does not enable authentication or authorization by default [4]. The database administrator can enable both authentication and authorization in the configuration file. When authorization is enabled, the database administrator can grant privileges either using role-based access control or user-based access control [4]. The HTTP interface for Cassandra, OpsCenter, supports authentication and authorization, but this is not enabled by default even if authentication and authorization is enabled in the Cassandra configuration file [7]. In order to extend authentication and authorization to OpsCenter, the database administrator must enable these protections through the OpsCenter [7].

3. APACHE CASSANDRA REMOTE CODEEXECUTION VULNERABILITIES

The default configuration in Apache Cassandra 1.2.0 through 1.2.19, 2.0.0 through 2.0.13, and 2.1.0 through 2.1.3 binds an unauthenticated JMX/RMI interface to all network interfaces, which allows remote attackers to execute arbitrary Java code via an RMI request.

4. LATEST ATTACKS ON CASSANDRA

The latest attack on Apache Cassandra is done by twitter user that goes by nickname of DunningKrugerEffect. During this attack empty table has been added to the database with name 'your_db_is_not_secure'. The purpose of this table is warn Cassandra user that their database can be very easily held for ransom in future if left online unprotected.[9]

5. SECURITY POLICIES IN APACHE CASSANDRA

Cassandra provides these security features to the open source community.

5.1 Client-To-Node Encryption

Cassandra includes an optional, secure form of communication from a client machine to a database cluster.

5.2 Authentication Based On Internally Controlled Login Accounts/Passwords

Administrators can create users who can be authenticated to Cassandra database clusters using the CREATE USER command. Internally, Cassandra manages user accounts and access to the database cluster using passwords.

5.3 Object Permission Management

Once authenticated into a database cluster using either internal authentication, the next security issue to be tackled is permission management. Authorization capabilities for Cassandra use the familiar GRANT/REVOKE security paradigm to manage object permissions.

5.4 SSL Encryption

A client and server are defined as two entities that are communicating with one another, either software or hardware. These entities must exchange information to set up trust between them. Each entity that will provide such information must have a generated key that consists of a private key that only the entity stores and a public key that can be exchanged with other entities. If the client wants to connect to the server, the client requests the secure connection and the server sends a certificate that includes its public key. The client checks the validity of the certificate by exchanging information with the server, which the server validates with its private key. If a two-way validation is desired, this process must be carried out in both directions. Private keys are stored in the keystore and public-keys are stored in the truststore.

5.5 Internal Authentication and Authorization

Internal authorization or internal authentication is based on Cassandra-controlled login accounts and passwords. Internal authentication works for the following clients when you provide a user name and password to start up the client.
Cassandra provides the familiar relational database GRANT/REVOKE paradigm to grant or revoke permissions to access Cassandra data. A super user grants initial permissions, and subsequently a user may or may not be given the permission to grant/revoke permissions. Object permission management is based on internal authorization.

5.6 Firewall Port Access

If you have a firewall running on the nodes in your Cassandra cluster, you must open up the following ports to allow bi-directional communication among the nodes, including certain Cassandra ports. If this isn't done, when you start Cassandra on a node, the node acts as a standalone database server rather than joining the database cluster.

6. CONCLUSION

This paper describes certain features of Cassandra that recommend Cassandra is most popular now a days and it is also open source so there is more possibility of vulnerabilities. Here we have list out some vulnerabilities in apache Cassandra and a latest attack. Here we have identified that the Cassandra's HTTP interface, OpsCenter can present vulnerabilities when used default configuration option. It can be mitigated by changing the network connection to bind to localhost or enable authentication and authorization.
Through many of the proposed Cassandra security policies the database can more secure from intruders and hackers.

7. REFERENCES

1. L. Okman, N. Gal-Oz, Y. Gonen, E. Gudes and J. Abramov, "Security issues inNoSQL databases," in *Trust, Security and Privacy in Computing andCommunications (TrustCom), 2011 IEEE 10th International Conference on*,

    a. 2011, pp. 541–547.

2. Tech Target:What is distributed database:Retrieved March 28,2017 from http://searchoracle.techtarget.com/definition/distributed-database

3. Datastax: Overview of Apache Cassandra: Retrieved March 29, 2017 from http://www.datastax.com/resources/tutorials/cassandra-overview.

4. *Apache Cassandra 2.2 for Windows Documentation*, Datastax Inc., 2015, pp. 8-97.

5. J. Han, E. Haihong, G. Le and J. Du, “Survey on NoSQL database,” in *Pervasive Computing and Applications (ICPCA), 2011 6th International Conference* on, 2011, pp. 363–366.

7. *CQL for Cassandra 2.x Documentation*, Datastax Inc., 2015, pp. 6-28.

8. *OpsCenter 5.2 User Guide Documentation*, Datastax Inc., 2015, pp. 8-78.

9. A. Zahid, R. Masood and M. Shibli, “Security of sharded NoSQL databases: A comparative analysis,” in *Information Assurance and Cyber Security (CIACS), 2014 Conference on, 2014, pp. 1–8.*

10. Catalin Cimpanu : A Benevolent Hacker Is Warning Owners of Unsecured Cassandra Databases, last update: 24 January,2017, Retrieved April 1,2017 from https://www.bleepingcomputer.com/news/security/a-benevolent-hacker-is-warning-owners-of-unsecured-cassandra-databases/