iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VII **Month of publication:** July 2020

DOI: https://doi.org/10.22214/ijraset.2020.30794

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VII July 2020- Available at www.ijraset.com

A Secured Method for accessing HDFS in Hadoop

Jolly Khurana[1], Dr. Kamlesh Sharma[2]
[1, 2]*Faculty of Technology, MRIIRS*

***Abstract:** Since Big data is so huge that it's become difficult to handle it, so it requires special technology which can handle bigdata. Hadoop is Apache Foundation's Framework which aims to provide efficient storage and analytics of big data; also, it is open system software. Two core technologies are associated with Hadoop i.e. HDFS and Map Reduce. HDFS is abbreviated for Hadoop Distributed File Technology, it's a special file system which provides efficient storage for big data in cluster of commodity hardware and based on stream access pattern. HDFS cluster Architecture is based on distributed file system therefore has Client server architecture. Since in HDFS cluster, there is no way to check the authenticity of client, therefore a method to incorporate Kerberos protocol in between Client and HDFS cluster is purposed to make the system secured. Kerberos is network authentication protocol which provides secured communication between client and server over unsecured network. Moreover, an Agent has been incorporated which is authorized to access the client's buffer and takes data out from it and loads it into HDFS cluster.*

***Keywords:** Big data, Hadoop, HDFS, Kerberos, Agent.*

I. INTRODUCTION

Data plays a vital role in everyone's life in some or the other way. In today's world where IT sector is growing rapidly, data management is one of the serios concerns. Managing data has become a challenging task now days. With the advancement in technology, many new software, tools and framework have been developed by the computer engineers in order to complete the work in less time and also to prevent data loss. Data warehouse and data mining came into picture that are used to store and analyse data. Such technologies work quite well when the data emerges in small unit like Kilobytes, megabytes and even Gigabytes. But such system fails when data starts overwhelming rapidly from various sources. This was the origin of Big data. Big data is the amount of data that is huge that it become difficult to handle using traditional data management techniques. With the advancement in digital sensors and communication technologies, enormous amount of data is generating which is useful in capturing valuable information for enterprises and business. This big data is hard to process using conventional technologies and require substantial parallel processing. So, there is a strong need for technology that are able to store and process Exabytes, Terabytes, Petabytes of data without tactually raising the cost of data warehousing.

In 2005, an open source project started by Doug Cutting, Mike Carmella and team who took the solution provided by Google which is developed by Apache. "It is a conventional stamp of the Apache software foundation. Hadoop framework is efficient enough to build applications which is capable to run on batch of computers and accomplish complete statistical analysis for enormous amount of data. A hadoop framework application functions in a condition that afford distributed storage and data processing across cluster of computers. Hadoop is constructed to scale up from single server to thousands of machines, each donating local computation and storage. So basically, Hadoop developed to store and carry out processing of Big data. The filesystem which is used to store big data is HDFS (Hadoop distributed Filesystem) and the technique that accomplish analytics of Big data is called Map Reduce.

II. HADOOP DISTRIBUTED FILESYSTEM(HDFS) AND ITS DRAWBACK

There are number of file system such as FS, HFTP FS and others which are supported but the most accepted file system is the Hadoop Distributed file system. The basic architecture of HDFS file system is like Google File system. HDFS provides the environment of distributed file system that is arrangement of thousands of computers to run on broad cluster of economical computer machines in an unfailing, fault tolerant fashion. HDFS works on master/slave architecture where master is the main node which comprised of a single Name Node (NN) that preserves the file system metadata and one or more slave Data Node (DN) that stores the physical data. A file disintegrate into several small blocks and those blocks are stored in a set of Data Node which are further documented in HDFS namespace. The read and write operation of the file system is concerned with Data Node (DN). They are also responsible for creation, deletion and reduplication of block on the basis of commands delivered by name Node (NN).

An HDFS client wanting to read a file first contact to NN for the location of data blocks comprising the file and then reads block contents from DN nearest to the client.

The client raised a request to the Name Node (NN) to select a group of Data Node (DN) to host the block replicas to writing the data. After that the client directly writes data to DN.

In the present architecture as show in figure 1. there is one Name Node for each cluster. The cluster can consist more than thousands of Data Node (DN) and tens of thousands of HDFS clients per cluster, as each DN may execute multiple application tasks concurrently.

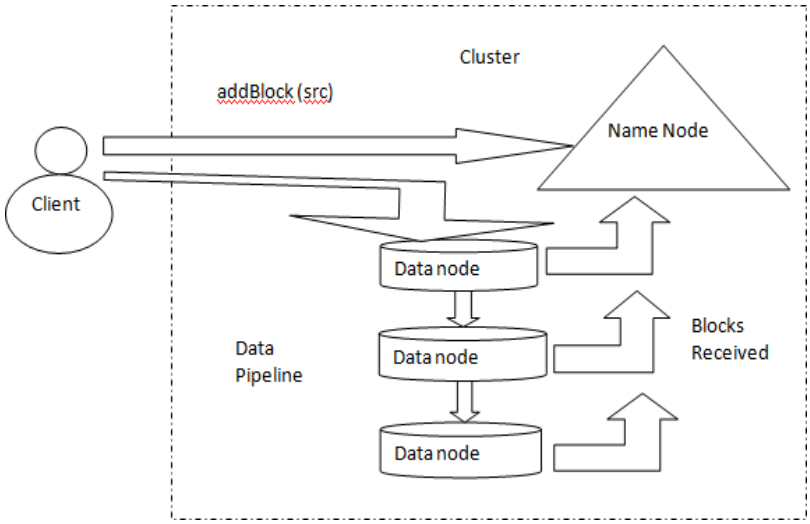


Fig 1. HDFS Cluster

Drawback-Now here the problem arises that there is no security of HDFS cluster as client is accessing the cluster directly. A client must be authorized and authenticated before accessing the HDFS cluster.

A protocol must be there that provide authenticity and authorization to client before accessing the cluster. Moreover, the client must be replaced by some agent that performs the task of client in order to reduce its work. An agent has to automatically loads the file of client into HDFS cluster or from HDFS cluster to client's machine. In this way these two problems can be solved. The protocol called Kerberos is incorporated between the client and HDFS cluster

III.AUTHENTICATION AND AUTHORIZATION BY KERBEROS

The secure communication over the unsecured network is done with the help of Kerberos network authentication protocol between the client and server. Kerberos was developed to securely identify their hosts. A three-party exchange is conducts between the client and server to prove the authentication of the client to the server. A ticket is presented by client to the server to prove its identity to the server. The ticket presented by the client is locked by client with the help of its secret key and unlocked by the server with the help of same key which is supports by Kerberos network protocol which support the symmetric cryptography. In this way the ticket remains locked over the network so that secure communication happened between the client and server.

A trusted third part Key Distribution Centre (KDC) issued tickets which used for secure communication over the network. The KDC, named by Needham and Schroeder, holds secret keys known by each client and server on the network. The client and server believe the authenticity of the tickets receives with the help of KDC. The validity of the ticket is for the finite time which is called the lifetime of the ticket when this time expires the ticket invalid. A new initiation is required to for later authentication with the help of KDC. The technologies are:

A. Realm

The authentication administrative domain is term as 'realm'. The main objective of the realm to provide an environment in which user is authentication is done by the server, host or service. It's not important that a user and a service must belong to the same realm. In multiple realm the different host are part of communication and developed a trust relationship between them, then the authentication can take place.

B. Principal

A principal is the name which is associated with each client, service and server, it's basically a unique identity like IP address. In KDC's Database, principals are stored instead of storing complete information of the hosts.

C. Ticket

A ticket is something that a person presents to another so as to urge entry somewhere, same is the case here; a client presents a ticket to an Application server in order to prove its legitimacy. Tickets are issued by the Authentication server and are encrypted using the secret key of the service they are intended for. Since this key have to secretly share solely between the authentication server and the server imparting the service, even the client is unknown of the content of the ticket. The prime detailsor information incorporated in a ticket are:

1) User's principal (for e.g. Username)
2) The information of the service it is intended for
3) The IP address of the client machine
4) The date and time (in timestamp format) when the tickets validity starts.
5) The ticket's maximum time to live.
6) The session keys

This is important as the authentication server doesn't have any control over an already issued ticket for a long time. The issuing of new tickets for certain user can be preventing with the help of realm administrator at any time. The preventing of the user who already issued the tickets is not done. This is force to integrate the security in the system with the help of limited lifetime or lifespan of the tickets.

D. Encryption

Since security is the main concern, encryption is utmost important to make secured communication. Kerberos supports symmetric key cryptography means a single key is used to encrypt and decrypt the message or packet or ticket to be transferred. Encryption is performed initially on the client side. Client encrypt its message then send it over the network to the server, server decrypts the message with the same key. On the other hand, server does the same work of encrypting its message and send to the client.

E. Key Distribution Centre (KDC)

A single physical server is dedicated for KDC is the center where exchanging of tickets actually take place. The KDC physical server then logically divided into three main parts.

a) KDC database
b) Authentication Server
c) Ticket Granting Server (TGS)

1) Database

a) The database is the container for entries associated with users and services i.e. principals. Each entry contains the following information:
b) To which Principal entry associated.
c) Lifetime of ticket associated with Principal.
d) If the lifetime of the ticket is increased then it has to be name again.
e) The behavior of the ticket is characterized by attributes or flags.
f) The expiration date of password.
g) The last expiration date of lifetime of the principal, after that no tickets will be issued.

2) Authentication Server (AS)

a) The Authentication Server which is also the part of KDC replies to the initial authentication request from the client, when the user, who is not authenticated yet, has to enter the password. In response to an authentication request, the AS issues a special ticket known as the Ticket Granting Ticket (TGT). If the user is actually who he says, he is authenticated to use.
b) The TGT to get other service tickets, without re- entering their password.

3) Ticket Granting Server (TGS)

a) The Ticket Granting Server is the KDC component which provides service tickets to clients with a valid TGT. Ticket Granting Server guarantees the authenticity of the client, moreover it also authorizes the user to access the service from the Application Server. and a service.

F. Session Key

Session keys are short lived private keys brought out by Kerberos. Client is known about this key and are used to encode the communication between the client and the server.

G. Authenticator

Authenticator is employed together with the ticket to prove that the client presenting a ticket is really the one it claims to be.

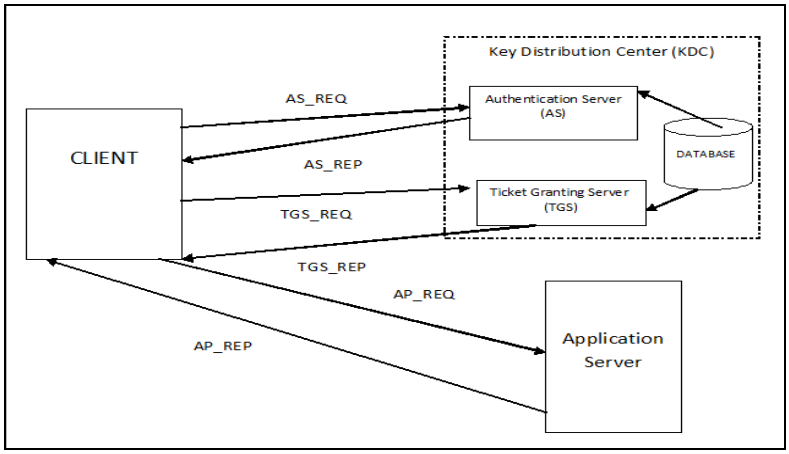


Fig. 2 Kerberos Architecture

1) Client login to his machine and make a request to KDC's Authentication Server to get ticket by providing his credentials.
2) KDC then checks client's information in its Database and if the information found then it replies back to client by providing TGT (Ticket granting ticket) which is encrypted by client's key as well as a session key.
3) Once the client gets the TGT, he can now make request to TGS by showing TGT. The request was made to access server.
4) KDC then gives response message to Client. The client received the response from the TGS. The response contained encrypted service ticket which is encrypted with the help of secret key and session key which generated by TGS.
5) Now after getting service ticket, client is able to request server to access the service. The request contains authenticator which is generated by client service session key.
6) Application Server then replies back to client by providing accessibility to its service after checking tickets.

IV. PROPOSED SYSTEM

In order to provide authenticity to the client, The system have added an Authentication protocol called KERBEROS in between Client and HDFS cluster, so that when client requests the Name node of the HDFS cluster to store its data, the Name node redirects the client towards Kerberos which authenticate the client and authorize the Agent. Agent after getting authority takes the client's data from client's buffer and put it into Data node. In this way, the whole system of storing big data into the HDFS cluster become more secure and also the client now doesn't have to loads its data into the Data node by itself.

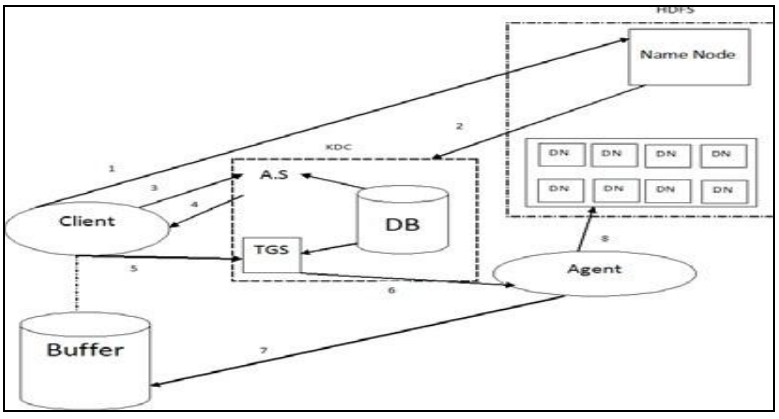


Fig 3. Proposed Architecture

V. STEPWISE STUDY OF A SYSTEM

1) Client first contacts to Name node of HDFS cluster in order to request to store its data.

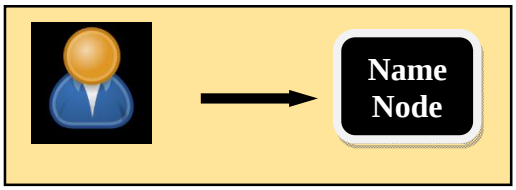


Fig4. Client to Name Node

2) Then, Name node being the Master node redirects the client's request to the Kerberos protocol which is placed in between client and HDFS cluster

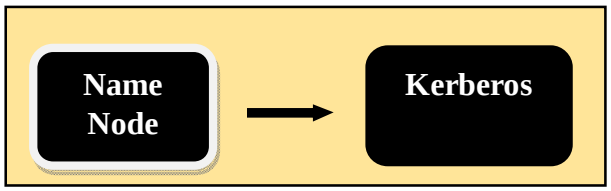


Fig5. Name Node redirects to Kerberos

3) In Kerberos, there is a Key Distribution centre where the client and server communicate. It consists of two servers and a database i.e.

a) Authentication Server (AS)
b) Ticket Granting Server (TGS)
c) Database

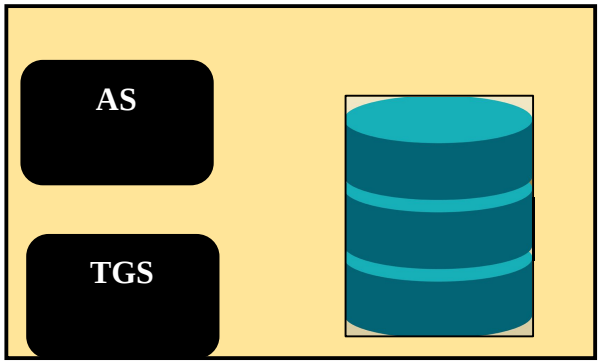


Fig6. Key Distribution Centre

4) The client then requests the Authentication server for the ticket which authenticates it i.e. TGT (Ticket Granting Ticket). This request is called Authentication Request (AS_REQ). The packet in which client sends the request is in the form;

AS_REQ = { Principalclient, Principalservice, IP_LIST, Lifetime}

$Principal_{client}$= Principal associated with the client $Principal_{service}$= Principal associated with the service that client wants

IP_LIST= IP address of the client machine Lifetime= Validity of the ticket issued

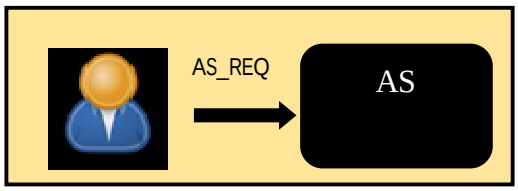


Fig7. Client Request for TGT

5) Then the Authentication Server checks the entry of Principal of client into the database. If the database contains client's identity or principal then it provides the ticket else send an error message to the client. The reply of the Authentication server is represented AS_REP. The encryption of TGT which send by AS is done with the help of TGS secret key with combination of Session key which is encrypted with the user's secret key and shared between Client and TGS. The packet is in the form;

SK= Session key Ku= user's key

$K_{TGS=}$ Secret Key of TGS

TGT= {$Principal_{client}$, $Principal_{service}$, IP_LIST, Timestamp, Lifetime, SK}

AS_REP= {Principal service, Timestamp, Lifetime, SK}Ku {TGT}K_{TGS}

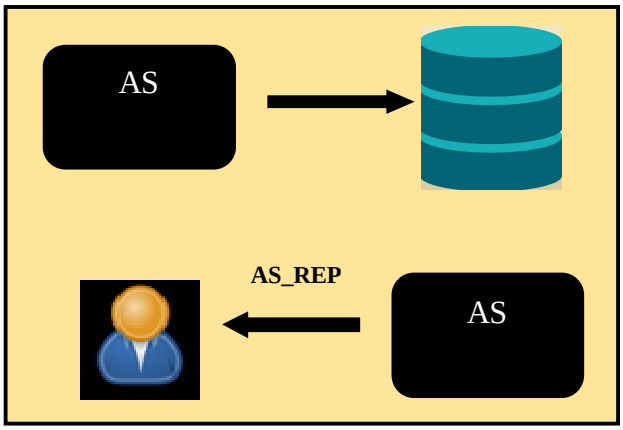


Fig8. Authentication Server Reply

6) The user asked by client to enter the password after getting the TGT. The string key function is applied after concatenated a string with password and with the client secret key a method is apply to decrypt message encrypted by the Authenticated server with the help of secret key of user stored in the KDC database. The authentic user is able to entered the correct password and able to extract the session from the TGT, which remains encrypted and stored in user`s credential cache

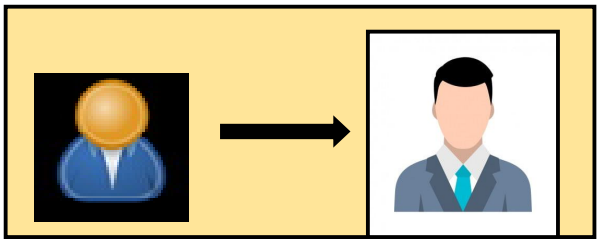


Fig9. Client Contacted User

7) The client thus send request to the TGS for the Service ticket, the message of the client is in form of packet contains Authenticator and TGT(encrypted with the KTGS). The request the Client k, Kerberos was implemented between client and to the TGS is represented TGS_REQ. The packet thus looks like;

Authenticator = { $Principal_{Client}$, Timestamp }SK

TGS_REQ = ($Principal_{Service}$,Lifetime, Authenticator) { TGT}KTGS

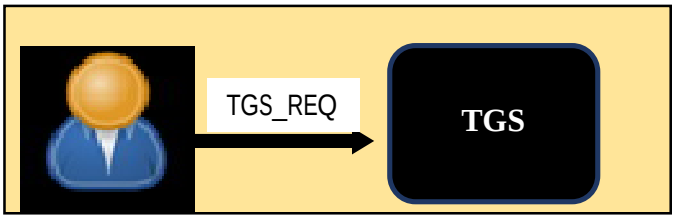


Fig10. Client Request for service ticket

8) The requested service principal ($Principal_{Service}$) verifies by the TGS and check the existence in the KDC database. The existence opens the TGT using the key and extracts the session key (SK) which it uses to decrypt the authenticator. It checks that TGT has not expired for the service ticket to be issued.

The $Principal_{Client}$ present in the authenticator matches the one present in the TGT etc.

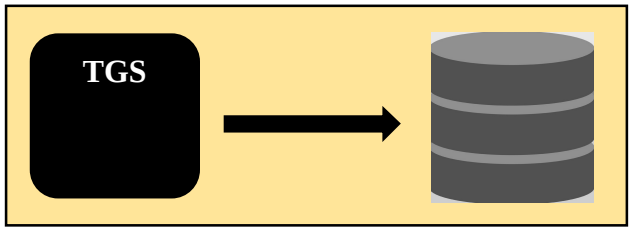


Fig11. TGS verifies TGT

9) TGT, instead of replying back to client i.e. providing service ticket to the client, gives this ticket to **AGENT,** since agent is a part of the server, has all the keys to decrypt the message and extract service ticket. Now Agent is authorized to take the data from user's buffer and put it into the Data node of HDFS cluster.

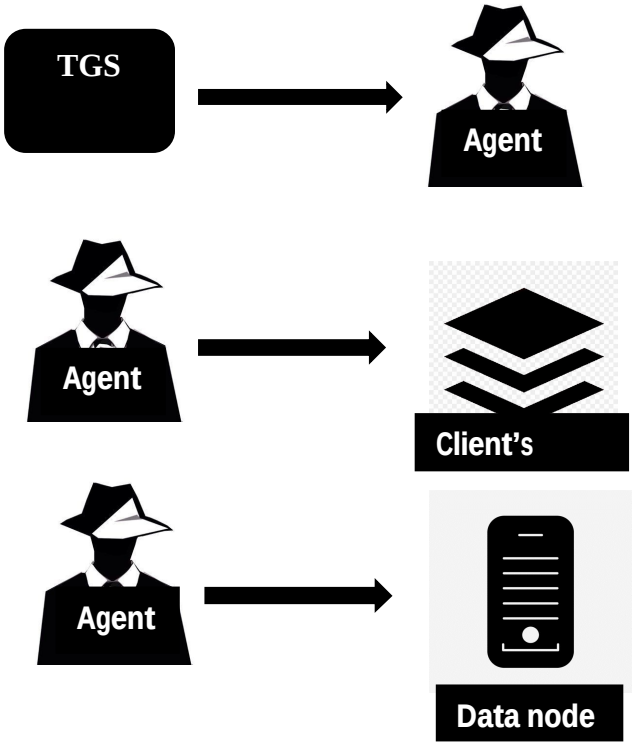


Fig12. TGS authorize Agent

VI. CONCLUSION

HDFS cluster. However, client is responsible to take its data from its buffer to Data node. This gave the motivation for further improvement in the security of HDFS cluster. This work improves the overall security of the system and makes it more reliable. This security yielded an improved version of HDFS cluster as well as changed the Kerberos architecture according to the need. This work though doesn't ensure the security in every situation, but surely illustrates the improvement in security which can be expected in the actual architecture of HDFS. This security improvement is although not too big, but is definitely provides a great method to secure the storage of Big Data into the Hadoop's HDFS cluster.

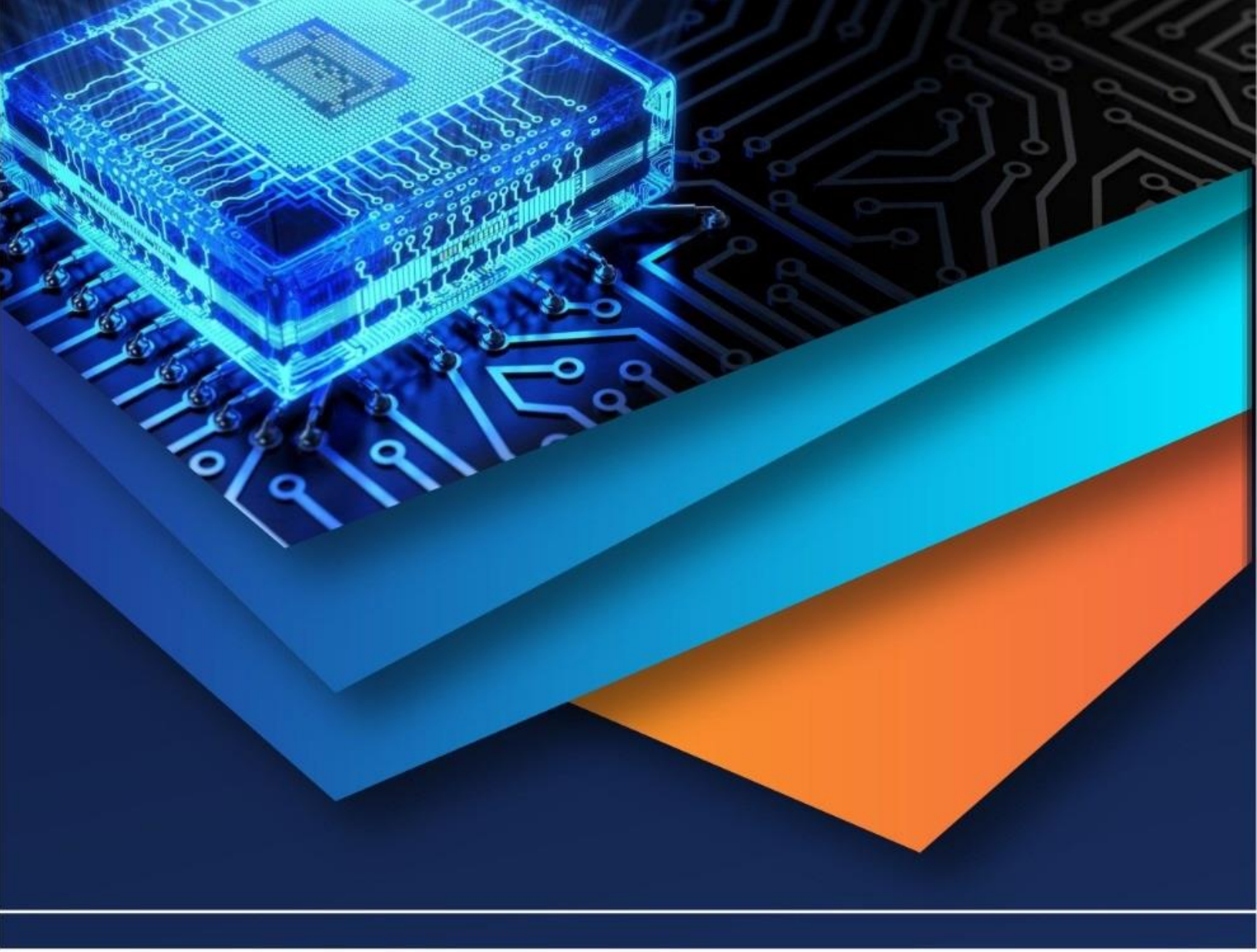
VII. FUTURE SCOPE

Storage of Big Data without losing a bit has been an issue since the origin of digital world. HDFS was invented to overcome the storage issue. Current infrastructure offers a comfortable environment for proper storage of data, but the structure in not that secured according to the requirement. The motive of this paper is to get an improved version of HDFS cluster by integrating Kerberos into it as well a slight change in the Kerberos Architecture.

The above proposed framework of HDFS cluster is implemented considering the available technology infrastructure. The process can be changed according to the network needs in the future. The framework is secure and scalable. But it requires some more modifications in the system since in Kerberos even an un- authenticated client can request for the ticket. This issue needs to be resolved in order to improve the security.

REFERENCES

[1] Neelam Singh, NehaGarg and Varsha Mittal, 'Big Data – insights, motivation and challenge', International Journal of Scientific & Engineering Research, Vol.4, Issue 12, December-2013.
[2] SushmaLakshkar, GeetKalani and VinodTodwal,'A Catholic Research on Big Data and Hadoop Environment', International Journal of Computer Applications, Vol. 130 – No.11, November2015.
[3] Stephen Kaisler, Frank Armour and J. Alberto Espinosa,'Big Data: Issues and Challenges Moving Forward', 46th Hawaii International Conference on System Sciences 2013.
[4] Harshawardhan S. Bhosale, Prof. Devendra and P. Gadekar, 'A Review Paper on Big Data and Hadoop, International Journal of Scientific and Research Publications, Vol. 4, Issue 10, October 2014 , Pg 1-7.
[5] Poonam S. Patil, Rajesh. N. Phursule, 'Survey Paper on Big Data Processing and Hadoop Components, International Journal of Science and Research (IJSR).
[6] Hadoop Tutorials available at http://www.tutorialspoint.com/hadoop/hadoop_tutorial.pdf.
[7] Jeffrey Shafer, Scott Rixner, and Alan L. Cox,'The Hadoop Distributed Filesystem: Balancing Portability and Performance'
[8] Konstantin Shvachko, HairongKuang, Sanjay Radia and RobertChansler,'The Hadoop Distributed File System'.
[9] Eman El-Emam, MagdyKoutb, HamdyKelash, and Osama Farag Allah, 'A Network Authentication Protocol Based on Kerberos', IJCSNS International Journal of Computer Science and Network Security, Vol.9 No.8, August 2009, pg 17-26.
[10] RandhirBhandari and Sachin Sharma, 'Kerberos: Simplified Ticketing', International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, Issue 11, November 2013.
[11] TraptiOzha, 'Kerberos: An Authentication Protocol', Int.J.Computer Technology &Applications,Vol 4, Issue 2, Pg 354-357.
[12] John T. Kohl, B. Clifford Neuman and Theodore Y. Ts'o, 'The Evolution of the Kerberos Authentication Service'.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY