International Journal on Emerging Technologies 11(1): 403-408(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

Data Authorization in Hadoop using Kerberos Authentication System and Transport Layer Security

Yallapragada Ravi Raju[1] and Haritha Donavalli[2]
[1]Assistant Professor, Department of Computer Science and Engineering,
Sri Vasavi Engineering College, Tadepalligudem (A.P.), India
Research Scholar, Department of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur (A.P.), India.
[2]Professor,Department of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur (A.P.), India.

(Corresponding author: Yallapragada Ravi Raju)
(Received 14 October 2019, Revised 17 December 2019, Accepted 24 December 2019)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: In the current scenario, a number of platforms are developed to build big data applications in both proprietary and open-source. One of the popular and easily accessible plat-form is Hadoop, which is an open source software for big-data processing. The prior version of Hadoop (CVE-2017-3162) did-not concentrated on security that lacks in secure authentication, which allows an unauthorized party to access the data. Currently, the authentication systems are utilized for user authentication in the Hadoop Distributed File System (HDFS), which is vulnerable to data-node hacking attacks and replay. In this research paper, Kerberos authentication system along with Transport Layer Security (TLS) encryption was proposed to protect the stored data in HDFS from replay and attacks. The proposed system permits HDFS clients to be authenticated by the data-node using block access token. The experimental result showed the efficiency and effectiveness of the proposed system on the basis of HDFS capacity, alerts across data-nodes, critical events across data-nodes, total bytes written across data-nodes, total blocks read across data-nodes, total blocks written across data-nodes, total transceivers across data-nodes, transceivers across data-nodes, send data packet transfer average time, average disk flush time across data-nodes, and import events and alerts are improved.

Keywords: Data authentication, Hadoop system, Hadoop distributed file system, Kerberos authentication system and Transport layer security.

Abbreviations: HDFS, Hadoop Distributed File System; TLS, Transport Layer Security; KDC, Key Distribution Center; TGT, Ticket-Granting Ticket; TGS, Ticket Granting Service, TGS; SSL, Secure Sockets Layer.

I. INTRODUCTION

In current decades, digital data are generated from many sources and the fast evolution of digital technologies increases the growth of big data. It delivers evolutionary break-through in several fields with the collection of large data-sets [1-2]. The main objective of the big-data analysis is to process the high velocity, veracity, volume, and variety of data using several computational and traditional intelligent techniques. Generally, big data refers to the collection of complex and large data-sets, which is hard to process using data processing applications and traditional data-base management tools [3-4]. Though, Hadoop system is developed to provide the solutions for massive datasets in applications like machine learning, real-time analytics, data mining, relational data analytics, and web analytics [5-6]. The Hadoop is an open-source, java-based, and scalable system for distributed large-scale processing. It scales from a single machine to multiple servers to set-up an execution plat-form with storage, high availability, and local computation [7-8]. Usually, the Hadoop system comprises of two major components; HDFS and MapReduce.

The HDFS contains geo-graphically dispersed data nodes, where the user data resides. Then, MapReduce is a programming model, which comprises of two essential tasks; map and reduce. Whereas, map converts the data into set of data, where the individual elements are sub-divided into tuples (value/key pairs). Respectively, reduce takes the output of map as an input and joins the data tuples (value/key pairs) into a smaller set of tuples [9]. Generally, the existing security methodologies for protecting virtualized infrastructures are categorized into two types such as, security analytics and malware detection and also the prior techniques are not efficient in attack environment [10]. Usually, malware detection comprises of two steps; monitoring hooks are placed at dissimilar points with-in the virtualized infrastructure and then update the database, which is used to identify the presence of attacks [11-12]. The origin of Hadoop system is very essential for preserving a basic idea about Hadoop security [13]. In this research paper, Kerberos authentication system was integrated with TLS encryption for protecting the data stored in HDFS from replay and attacks.

This paper is prearranged as follows. Section II surveys several existing research papers in Hadoop security system. Section III shows problem statement of existing methodologies. Section IV details theoretical explanation and quantitative analysis of the proposed system. Conclusion is made in section V.

II. LITERATURE REVIEW

Many new approaches are developed by the researchers in Hadoop security system. In this segment, a brief evaluation of a few essential contributions to the existing literature papers are presented.

Jeong and Kim (2015) developed a new authentication approach: token based authentication in HDFS for protecting the sensitive data from impersonation and replay attacks. Here, the developed approach permits HDFS clients to be authenticated by the data node using block access token. Usually, the existing HDFS authentication protocols adopts only public key exchange methodologies, but the developed approach established hash chain of keys for HDFS authentication. In the experimental section, the developed approach performance was evaluated in light of area efficiency, communication power and computing power. In this paper, an external encryption algorithm was required to protect the privacy of enterprise and personal data stored in HDFS [14].

Jeong *et al.*, (2016) presented a new protocol for public key encryption on the basis of hash chain approach instead of using the key exchange approaches. In the developed protocol, HDFS blocks in the data nodes were accessed by using a valid block-access token. In the experimental phase, the developed protocol performance was evaluated by means of area efficiency, communication power, and computational power. Related to other conventional HDFS systems, the developed protocol delivers better performance. The way to protect user privacy in HDFS data contains personally identifiable information within the organization, which was further needed to be studied in this research paper [15].

Win *et al.*, (2017) developed a new methodology for big data security. In this research paper, user application logs and network logs were collected from the guest virtual machines, which were stored in the HDFS. Then, Map-Reduce parser and graph based event correlations were applied to extract the attack features. Besides, determine the presence of attack using logistic regression and belief propagation approaches. Here, logistic regression method was used to calculate the conditional probability attacks on the basis of input attributes and belief propagation approach was used to evaluate the existence of attacks. The experimental result shows the efficiency and effectiveness of the developed methodology on the basis of communication and computational power. Though, the developed methodology provides insecure data sharing in the condition of maximum overhead [16].

Johri *et al.*, (2018) presented two applications that disseminates cryptography between the MapReduce jobs. The first application was encrypting the input data, which were resides in HDFS and the second application was managing the decryption of encrypted data. The experimental result shows the comparison between the two cryptographic algorithms. As data was encrypted in HDFS, the developed system does not worry about where the data resided. Privacy-preserving in Hadoop provides an option to securely access and manage the data that was cost-effective and scalable. Additionally, password-based encryption was used to decrease the overhead of key management. It could be only implemented in multi node Hadoop setup with larger files and more cryptography algorithms [17].

Shetty *et al.*, (2019) developed a multi-layer policy-based access control approach for Hadoop eco-system that depends on the concept of role-based access control system. The developed approach effectively prevents the un-authorized access for cluster resources and inflicts access control for data-owners. Still, the developed approach needs to improve the role-based access control system for dynamic permission management, securing the web communication channels, multiple applications under one eco-system, integrating with native portable operating system interface permissions, and real-time activity monitoring of all users [18].

III. PROBLEMS IN HADOOP SYSTEM

This section describes the problems faced by the users in Hadoop security and authentication and also detailed the solutions to the described problems.

– Unsecure network transport.

– In a few circumstances, authorized persons are not allowed to access sensitive data.

– User verification is necessary for map reduce code execution, because sometimes malicious users can also submit a job.

– No message-level security.

Research Gap: To address the above-mentioned concerns and also to improve the data security in Hadoop system, the following points needed to be considered.

– Need to protect the data (both at transit and rest state) using an encryption method.

– Ensure that authorized Hadoop users is only accessing the data.

– Ensure that data-access history for all users are recorded on the basis of compliance regulations.

IV. PROPOSED SYSTEM

The proposed Hadoop security system consists of four phases; enable Hadoop security utilizing Kerberos, select HDFS service, select category: security, and enable data transfer encryption by applying TLS. The workflow of the proposed Hadoop security system is graphically denoted in Fig. 1.

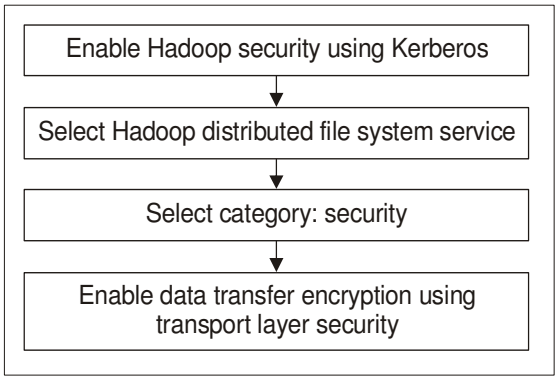

Fig. 1. Work flow of proposed Hadoop security system.

A. Kerberos authentication system

In the proposed system, enable the Hadoop security using Kerberos authentication system. This authentication system uses conventional shared secret cryptography for preventing or securing the data packets from replay attacks and eaves-dropping. The

basic steps involved in Kerberos environment are listed below:

Case I: Client request an authentication ticket from Key Distribution Center (KDC).

– Initially, KDC validates the authorization and then it provides the session key and Ticket-Granting Ticket (TGT) to the authorized person.

– Here, the TGT is encrypted by utilizing the Ticket Granting Service (TGS) secret key.

– Finally, the client stores the TGT. If it expires, the local session manager will request for another TGT.

Case II: Client request to access other resource or service on the network.

– Initially, client sends the current TGT to the TGS using service principal name.

– Then, KDC validates the user TGT for accessing the service.

– TGS sends session key to the client.

– Finally, the client sends session key to the service for accessing other resource or service on the network.

The Kerberos authentication system allows only the authorized users of HDFS for accessing and transferring the files among different nodes. The general block diagram of Kerberos authentication system is given in Fig. 2.

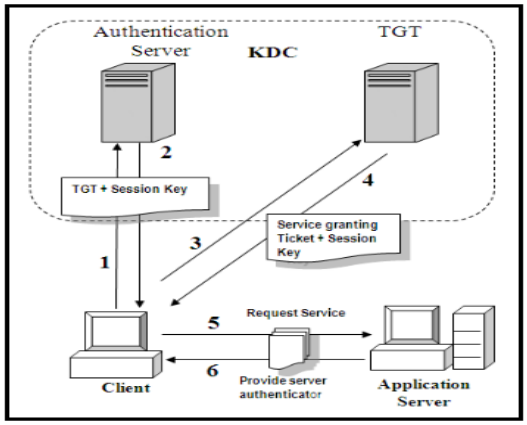

Fig. 2. General block diagram of Kerberos authentication system.

B. Hadoop distributed file system

The HDFS is a data storage system, which is mainly used in Hadoop based applications. Usually, the HDFS splits the large input data into small parts that are handled or managed by dissimilar machines in the cluster, which is given in the Fig. 3. In addition, the HDFS utilized data-node and named-node architectures to implement a distributed file system that delivers high performance data access across highly scalable Hadoop clusters.

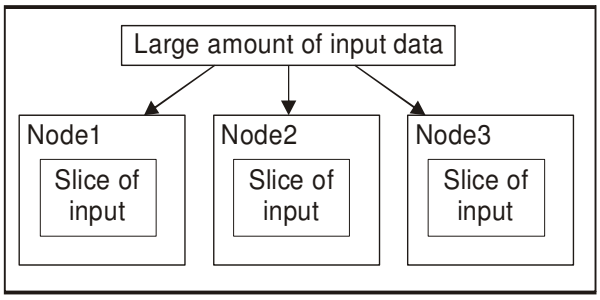

Fig. 3. Data distribution across nodes at load time.

The major components in HDFS are; name-node, data-node, secondary name-node, data replication, and rack awareness. The detailed explanation about the HDFS components are described below. Fig. 4 denotes the graphical representation of HDFS.

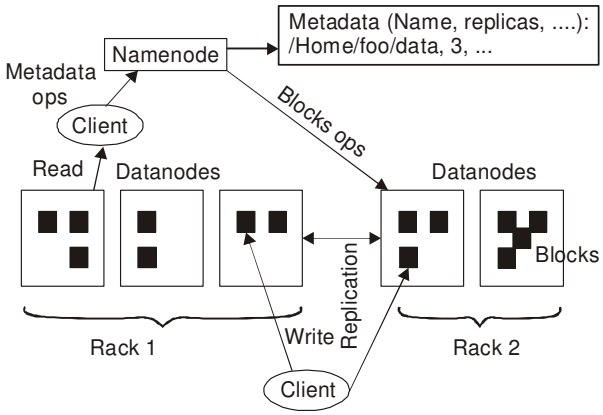

Fig. 4. General architecture of HDFS.

(a) Name-node: In HDFS architecture, name-node is a master-node that manages and handles the blocks present in the data-nodes. The major two operations of name-node are; control access to the files by the client and managing the file system namespace. The name-node also records the meta-data of all files stored in the clusters, for instance, permissions, location of stored blocks, file size, hierarchy, etc.

(b) Data-node: The data-node is commodity hardware in HDFS architecture that comprises of data-node software and Linux/GNU operating system in a cluster. As per the client request, the data-node performs read and write operations in the file system. The data-nodes also performs the operations like replication, creation and deletion on the basis of name-node instructions.

(c) Secondary name-node: The major role of secondary name-node in HDFS architecture is to merge the editlog and FSImage that helps to reduce the system memory space.

(d) Data replication: The replication management in HDFS architecture delivers a reliable way to store an enormous amount of data in a distributed environment as data-blocks. The data-blocks are also simulated to deliver fault tolerance in HDFS. The default replication factor in HDFS is three, which is configurable.

(e) Rack awareness: In Hadoop system, rack awareness is the concept of selecting the data-nodes on the basis of rack information. The rack awareness is essential to enhance the cluster performance, to improve the network band-width, and data reliability and availability.

C. Transport layer security

In this sub-section, TLS is applied for enabling the data transfer encryption. TLS is a protocol that delivers data integrity and privacy between the two communication applications. Currently, TLS is widely used as a security protocol in web browsers for secure file transfer, voice over internet protocol, instant messaging, etc. The TLS protocol is developed from the Netscape's Secure Sockets Layer (SSL) protocol. Compared to SSL, TLS is a more effective and secure protocol for key material generation, and message authentication.

Generally, the TLS comprises of two layers (TLS handshake protocol and TLS record protocol) based on the protocol specification.
The handshake protocol allows the client and server to authenticate each other and also used as a cryptographic keys and encryption algorithm for data exchange. Besides, the record protocol delivers connection security between the communication applications. The TLS record protocol should be on top of a transport layer and the standard specifies for four higher-level protocols are represented in Fig. 5.

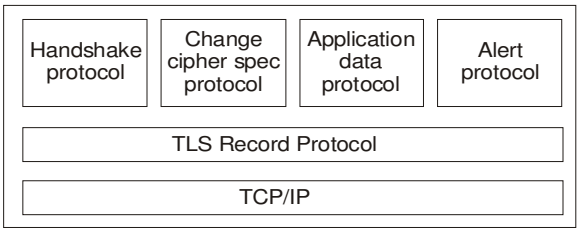

Fig. 5. TLS record layers.

D. Quantitative evaluation
The proposed system was simulated by using Hadoop cloudera 5.13 with 3.5 GHz Intel i5 processor, 4 GB RAM and 1 TB hard disc. In addition, a few existing hadoop systems available in the market are represented in Table 1.

Table 1: Existing hadoop system.

Hadoop system	
CVE-2018-11767	Hadoop 2.7.5 to 2.9.1
CVE-2018-11766	Apache hadoop 2.7.4 to 2.7.6
CVE-2017-7669	Apache hadoop 2.8.0
CVE-2017-3162	Apache hadoop before 2.7.0
CVE-2017-3161	Apache hadoop before 2.7.0
CVE-2017-15713	Apache hadoop 3.0.0

The performance of the proposed system was evaluated in light of HDFS capacity, alerts across data-nodes, critical events across data-nodes, total bytes written across data-nodes, total blocks read across data-nodes, total blocks written across data-nodes, total transceivers across data-nodes, transceivers across data-nodes, send data packet transfer average time, average disk flush time across data-nodes, and import events and alerts. The Figs. 6, 7 and 8 represents the performance evaluation of proposed system in terms of HDFS capacity, alerts across data-nodes, and critical events across data-nodes.

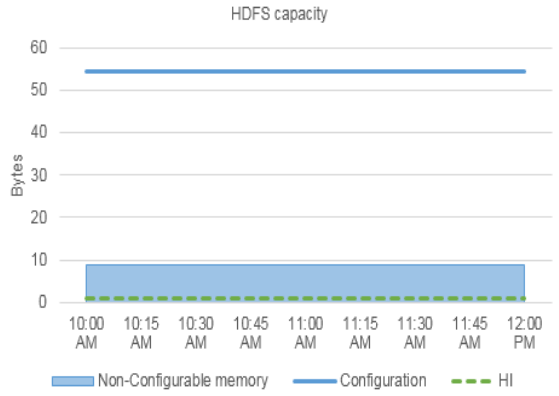

Fig. 6. Performance evaluation of proposed system in terms of HDFS capacity.

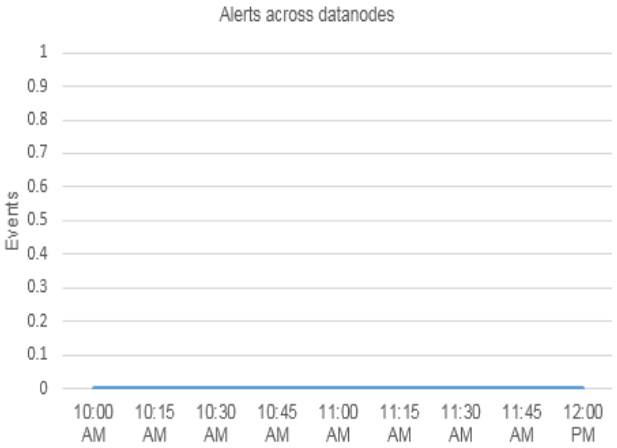

Fig. 7. Performance evaluation of proposed system in terms of alerts across data-nodes.

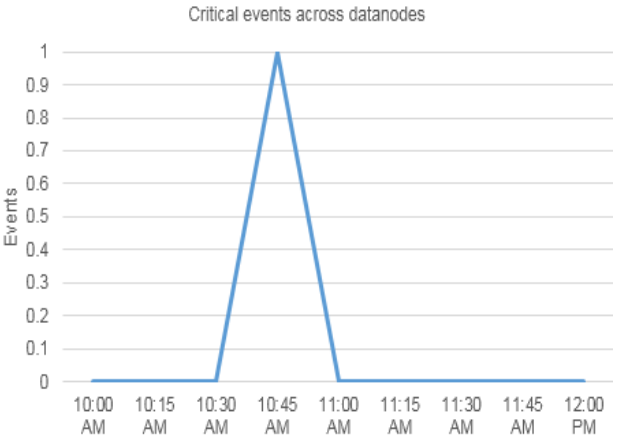

Fig. 8. Performance evaluation of proposed system in terms of critical events across data-nodes.

Figs. 9, 10, 11, 12 and 13 represents the performance evaluation of proposed system by means of total bytes written across data-nodes, total blocks read across data-nodes, total blocks written across data-nodes, total transceivers across data-nodes, and transceivers across data-nodes.

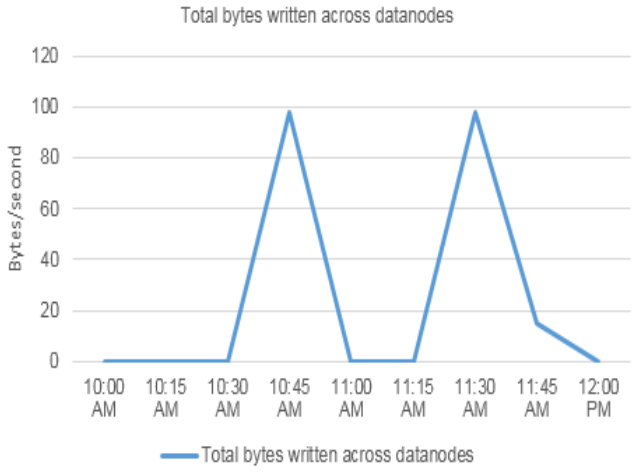

Fig. 9. Performance evaluation of proposed system in terms of total bytes written across data-nodes.

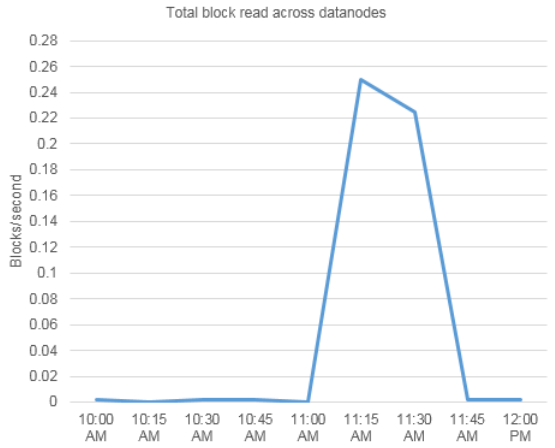

Fig. 10. Performance evaluation of proposed system in terms of total blocks read across data-nodes.

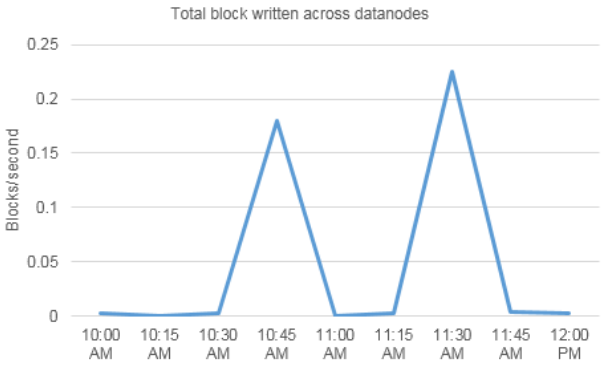

Fig. 11. Performance evaluation of proposed system in terms of total blocks written across data-nodes.

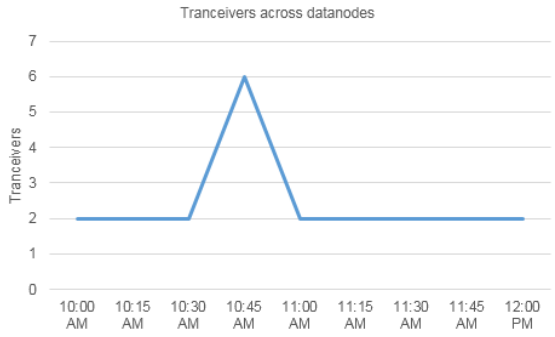

Fig. 12. Performance evaluation of proposed system in terms of total transceivers across data-nodes.

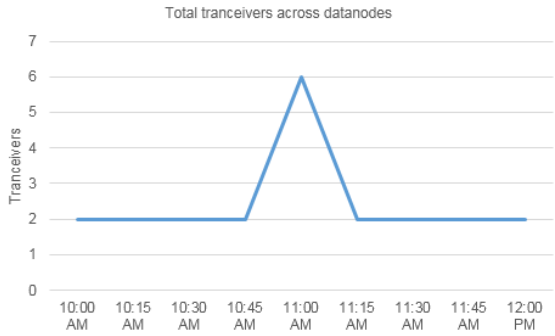

Fig. 13. Performance evaluation of proposed system in terms of total transceivers across data-nodes.

Figs. 14, 15 and 16 represents the performance evaluation of proposed system by means of send data packet transfer average time, average disk flush time across data-nodes and import events and alerts.

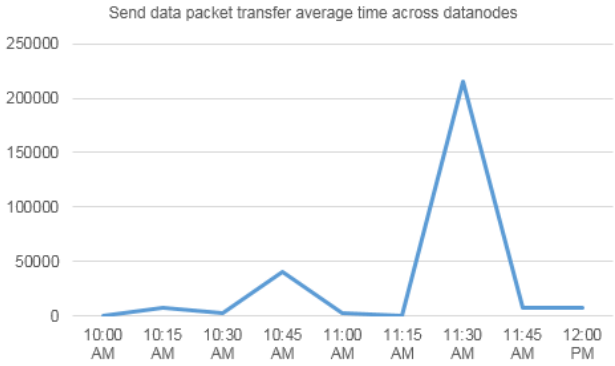

Fig. 14. Performance evaluation of proposed system in terms of send data packet transfer average time.

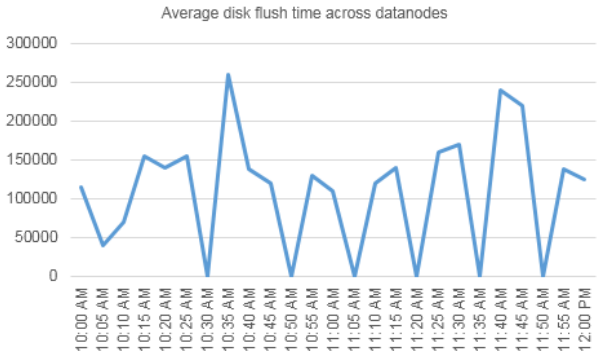

Fig. 15. Performance evaluation of proposed system in terms of average disk flush time across data-nodes.

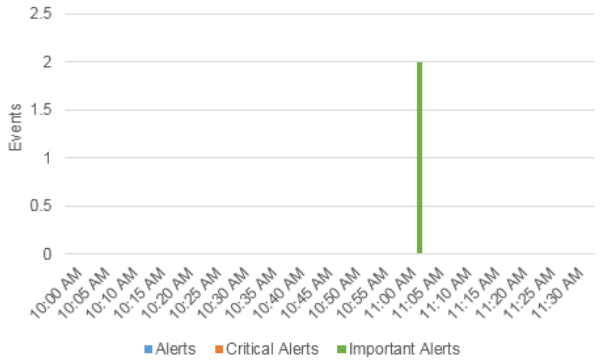

Fig. 16. Performance evaluation of proposed system in terms of import events and alerts.

V. CONCLUSION

In recent periods, Hadoop is the most popular platform for processing big-data, because it includes the advantages like fast speed, low costs and easy convenience. Nowadays, Hadoop is extensively utilized in private and government sectors, where its security is considered to be a major issue. In this research paper, a new authentication system was proposed for the clients in order to analyze the data security problems in the Hadoop system. The proposed system utilized TLS encryption along with Kerberos authentication system for protecting the data that stored in HDFS from replay and data attacks. Usually, the Kerberos protocol utilizes

a specific ticketing system, which provides faster authentication related to other protocols. Compared to other existing security systems in Hadoop, the proposed system showed an effective performance.

VI. FUTURE SCOPE

In future work, a new authentication system was developed with the latest version of Hadoop to further improve the data security.

Conflict of Interest. No.

REFERENCES

[1]. Shehzad, D., Khan, Z., Dağ, H., & Bozkuş, Z. (2016). A novel hybrid encryption scheme to ensure Hadoop based cloud data security. *International Journal of Computer Science and Information Security (IJCSIS)*, *14*(4).

[2]. Thakur, P. K., Kumar, R., Ali, R., & kumar Malviya, R. (2011). A new approach of bully election algorithm for distributed computing. *International Journal of Electrical, Electronics and Computer Engineering (IJEECE), 1*(1), 72-79.

[3]. Kanyeba, M., & Yu, L. (2016). Securing Authentication Within Hadoop. In 2016 *International Conference on Electrical, Mechanical and Industrial Engineering. Atlantis Press*, 100-103.

[4]. Karn, R., Kumar, Y., & Agnihotri, G. (2011). Development of ACO algorithm for service restoration in distribution system. *International Journal on Emerging Technologies*, *2*(1), 71-77.

[5]. Chattaraj, D., Sarma, M., Das, A. K., Kumar, N., Rodrigues, J. J., & Park, Y. (2018). HEAP: An Efficient and Fault-Tolerant Authentication and Key Exchange Protocol for Hadoop-Assisted Big Data Platform. *IEEE Access*, *6*, 75342-75382.

[6]. Gandhi, S. K., & Kumar, T. P. (2012). Election Administration Algorithm for Distributed Computing. *International Journal of Electrical, Electronics and Computer Engineering*, *1*(2), 1-6.

[7]. Zheng, K., & Jiang, W. (2014, October). A token authentication solution for hadoop based on kerberos pre-authentication. In 2014 *International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 354-360). IEEE.

[8]. Khalil, I., Dou, Z., & Khreishah, A. (2014). TPM-based authentication mechanism for apache hadoop. In *International Conference on Security and Privacy in Communication Networks* (pp. 105-122). Springer, Cham.

[9]. Somu, N., Gangaa, A., & Sriram, V. S. (2014). Authentication service in hadoop using one time pad. *Indian Journal of Science and Technology*, *7*(4), 56-62.

[10]. Parmar, R. R., Roy, S., Bhattacharyya, D., Bandyopadhyay, S. K., & Kim, T. H. (2017). Large-scale encryption in the Hadoop environment: Challenges and solutions. *IEEE Access*, *5*, 7156-7163.

[11]. Rahul, P. K., & GireeshKumar, T. (2015). A novel authentication framework for Hadoop. In *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems* (pp. 333-340). Springer, New Delhi.

[12]. Fu, X., Gao, Y., Luo, B., Du, X., & Guizani, M. (2017). Security threats to Hadoop: Data leakage attacks and investigation. *IEEE Network*, *31*(2), 67-71.

[13]. Shetty, M.M., & Manjaiah, D.H. (2016). New Security Architecture for Big Data Hadoop. In *International Conference on Emerging Research in Computing, Information, Communication and Applications*, (pp. 469-480). Springer, Singapore.

[14]. Jeong, Y. S., & Kim, Y. T. (2015). A token-based authentication security scheme for Hadoop distributed file system using elliptic curve cryptography. *Journal of Computer Virology and Hacking Techniques*, *11*(3), 137-142.

[15]. Jeong, Y. S., Shin, S. S., & Han, K. H. (2016). High-dimentional data authentication protocol based on hash chain for Hadoop systems. *Cluster Computing*, *19*(1), 475-484.

[16]. Win, T. Y., Tianfield, H., & Mair, Q. (2017). Big data based security analytics for protecting virtualized infrastructures in cloud computing. *IEEE Transactions on Big Data*, *4*(1), 11-25.

[17]. Johri, P., Arora, S., & Kumar, M. (2018). Privacy Preserve Hadoop (PPH)—An Implementation of BIG DATA Security by Hadoop with Encrypted HDFS. In *Information and Communication Technology for Sustainable Development* (pp. 339-346). Springer, Singapore.

[18]. Shetty, M. M., Manjaiah, D. H., & Hemdan, E. E. D. (2019). Policy-Based Access Control Scheme for Securing Hadoop Ecosystem. In *Data Management, Analytics and Innovation* (pp. 167-176). Springer, Singapore.

How to cite this article: Ravi Raju, Yallapragada and Donavalli, Haritha (2020). Data Authorization in Hadoop using Kerberos Authentication System and Transport Layer Security. *International Journal on Emerging Technologies*, *11*(1): 403–408.