A Survey on Security of Hadoop

Masoumeh RezaeiJam
Department of Computer Engineering
University of Tabriz
Tabriz, Iran
m_rezaeijam90@ms.tabrizu.ac.ir

Leili Mohammad Khanli
Department of Computer Engineering
University of Tabriz
Tabriz, Iran
l-khanli@tabrizu.ac.ir

Mohammad Kazem Akbari
Department of Computer Engineering and IT
Amirkabir University of Technology
Tehran, Iran
akbarif@aut.ac.ir

Morteza Sargolzaei Javan
Department of Computer Engineering and IT
Amirkabir University of Technology,
Tehran, IRAN
msjavan@aut.ac.ir

***Abstract*—Trusted computing and security of services is one of the most challenging topics today and is the cloud computing's core technology that is currently the focus of international IT universe. Hadoop, as an open-source cloud computing and big data framework, is increasingly used in the business world, while the weakness of security mechanism now becomes one of the main problems obstructing its development. This paper first describes the hadoop project and its present security mechanisms, then analyzes the security problems and risks of it, pondering some methods to enhance its trust and security and finally based on previous descriptions, concludes Hadoop's security challenges.**

Keywords-Security; Trust; Hadoop; BigData; MapReduce; Cloud Computing

I. INTRODUCTION

Today, data explosion is a reality of digital universe and the amount of data extremely increases even in every second. IDC's latest statistics show that rate of structured data in the Internet now have been grown about 32%, and unstructured data about 63%. To 2012, the unstructured data will occupies more than 75% proportion of the entire amount of data in the Internet [1]. The volume of digital content of the world grows to 8ZB by 2015 [2]. One common programming model to handle and process these extreme amount of Big Data is MapReduce [3]. Apache Hadoop [4] is an open-source software framework and well-known implementation of MapReduce model that supports data-intensive distributed applications. Hadoop Distributed File System (HDFS) is a distributed, scalable, and portable file system written in Java for the Hadoop framework, and actually it is cloud storage the most widely used tool [5]. In fact in the next 5 years, 50 percent of Big Data projects expect to be run on Hadoop. Financial organizations using Hadoop started to store their confidential sensitive data on Hadoop clusters. So, a need for a strong authentication and authorization mechanism to protect the sensitive data is observed and also there is a need for a highly secure authentication system to restrict the access to the confidential business data that are processed and stored in an open framework like Hadoop [6].

However, whether the user is assured that put the private data to various clouds is the key to widely promotion of cloud storage. Initially, Hadoop had no security framework and it considers that the entire cluster, user and the environment were trusted. Even though it had some authorization controls like file access permissions, a malicious user can easily impersonate a trusted used as the authentication were on the basis of Password. Later on, Hadoop cluster moved on to private networks, where the users have equal rights to access the data stored in the cluster [6, 7].

Equal access to all users gives the malicious user the possibility of firstly, read or modify the data in the other's cluster and secondly, suppress or kill the other job to execute his job earlier than the other to complete job from a malicizous user because the data node does not enforce access control policies [6, 7].

The rest of the article will be as follows: In Section 2 we details about Hadoop project and its present security level and threats. Then Section 3 briefs about Apache Sentry, Sections 4, 5, 6 and 7 explain about some existing mechanisms and methods proposed to make Hadoop cluster more secure and finally Section 8 summaries the paper.

II. APACHE HADOOP PROJECT

The Apache Hadoop project develops open-source software for reliable, scalable, distributed computing. It is a framework that by the use of plain programming models, permits for the distributed parallel processing of big data sets in the size of petabytes and exabytes across clusters of computers so that a cluster of hadoop can easily scale out and also scale up from single servers to thousands of machines which each of them offer local computation and storage. Many companies like amazon, facebook, yahoo, etc. store and process their data on hadoop that proves its popularity and robustness. The library itself is designed to detect and handle failures at the application layer, rather than rely on hardware to deliver high-availability, so delivering a highly-available service on the upside of a cluster of computers, each of which may be prone to failures [5][8].

978-1-4799-5487-2/14/$31.00 ©2014 IEEE

However, enterprises wants to protect sensitive data, while the weakness of security mechanism now becomes one of the main problems obstructing hadoop's development and use [9] and because of the lack of a valid user authentication and data security defense measures, Hadoop is now facing many security problems in the data storage [3].

A. *Present Hadoop Security Level*

Hadoop default means consider network as trusted and hadoop client uses local username. In default method, there is no encryption between hadoop and client host [10] and in HDFS, all files are stored in clear text and controlled by a central server called NameNode. So, HDFS has no security appliance against storage servers that may peep at data content. Additionally, Hadoop and HDFS have no strong security model, in particular the communication between datanodes and between clients and datanodes is not encrypted [11]. To solve these problems, some mechanisms have been added to Hadoop to maintain them. For instance, by strong authentication, hadoop is secured with Kerberos and thorough it, provides mutual authentication and protects against eavesdropping and replay attacks. Every user and service has a Kerberos "principal" and credentials are by Service: keytab [1] s and User: password which RPC [2] Encryption should be enabled [10].

Layers of defense for a hadoop cluster are [12, 13]

1. Perimeter Level Security : Network Security firewalls, Apache Knox gateway
2. Authentication : Kerberos
3. Authorization : e.g. HDFS permissions, HDFS ACL[3]s, MR ACLs
4. OS Security and data protection : encryption of data in network and HDFS

Now we provide description for these layers as follow.

1) *Apache Knox Gateway*

The Apache Knox Gateway [14] is a system that provides a single point of authentication and access for Apache Hadoop services. It accesses over HTTP/HTTPs to Hadoop Cluster and provides the following features [12]:

- Single REST API Access Point
- Centralized authentication, authorization and audit for Hadoop REST/HTTP services
- LDAP [4] /AD [5] Authentication, Service Authorization and Audit
- Eliminates SSH edge node risks
- Hides Network Topology.

2) *Authentication*

Authentication means to identify who you are. Providers with the role of authentication are responsible for collecting credentials presented by the API consumer, validating them and communicating the successful or failed authentication to the client or the rest of the provider chain [14]. By this primer security, untrusted users do not have access to the cluster network and trusted network, everyone is good citizen. Your identity is determined by client host.

For strong authentication, Hadoop uses [15]

- Kerberos
- LDAP, ActiveDirectory
- LDAP, AD integrated with Kerberos, establishing a single point of truth
- Single point of truth

Kerberos is a computer network authentication protocol which works on the basis of "tickets" to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner [3].

In the development of HDFS cluster, it places the trusted server authentication key in each node of the cluster to achieve the reliability of the Hadoop cluster node communication, which can effectively prevent non-trusted machines posing as internal nodes registered to the NameNode and then process data on HDFS. This mechanism is used throughout the cluster. So from storage perspective, Kerberos can guarantee the credibility of the nodes in HDFS cluster [3].

Kerberos can be connected to corporate LDAP environments to centrally provision user information. Hadoop also provides perimeter authentication through Apache Knox for REST APIs and Web services [16].

Hadoop relies completely on Kerberos for authentication between the client and the server. Hadoop 1.0.0 version comes with the Kerberos mechanism. An encrypted token the authentication agent will be requested by the client. Using this, he can request for a particular service from the server [3, 6].

However, Kerberos is ineffective against Password guessing attacks and does not provide multipart authentication [6].

3) *Authorization*

Authorization or entitlement is the process of ensuring that users have access only to data as per corporate policies. Hadoop already provides fine-grained authorization via file permissions in HDFS, resource-level access control for YARN [17] and MapReduce [18], and coarser-grained access control at a service level [16].

[1] Encrypted key for servers (similar to "password") which is generated by server such as Kerberos or Active Directory [10] B. Noland. (2013). *6 ways to exploit Hive and what to do about*. Available: http://www.slideshare.net/cloudera/deploying-enterprisegrade-security-for-hadoop

[2] Remote procedure call

[3] Access Control List

[4] Lightweight Directory Access Protocol

[5] Active Directory

The authorization role is used by providers that make access decisions for the requested resources based on the effective user identity context. This identity context is determined by the authentication provider and the identity assertion provider mapping rules. Evaluation of the identity contexts user and group principals against a set of access policies is done by the authorization provider in order to determine whether access should be granted to the effective user for the requested resource [14].

Out of the box, the Knox Gateway provides an ACL based authorization provider that evaluates rules that comprise of username, groups and ip addresses. These ACLs are bound to and protect resources at the service level. That is, they protect access to the Hadoop services themselves based on user, group and remote ip address [14].

To provide a common authorization framework for the Hadoop platform, providing security administrators with a single administrative console to manage all the authorization policies for Hadoop components is the goal of Hadoop's developers [16].

4) OS Security and Data Protection

Data protection involves protecting data at rest and in motion, including encryption and masking. Encryption provides an added layer of security by protecting data when it is transferred and when it is stored (at rest), while masking capabilities enable security administrators to desensitize PII for display or temporary storage. In Hadoop it will be continued to leverage the existing capabilities for encrypting data in flight, while bringing forward partner solutions for encrypting data at rest, data discovery, and data masking [16].

B. Security Threats in Hadoop

Hadoop does not follow any classic interaction model as the file system is partitioned and the data resides in clusters at different points. One of the two situations can happen: job runs on another node different from the node where the user is authenticated or different set of jobs can run on a same node. The areas of security breach in Hadoop are

i. Unauthorized user can access the HDFS file

ii. Unauthorized user can read/write the data block

iii. Unauthorized user can submit a job, change the priority, or delete the job in the queue.

iv. A running task can access the data of other task through operating system interfaces

Some of the possible solutions can be

- Access control at the file system level.
- Access control checks at the beginning of read and write
- Secure way of user authentication

Authorization is the process of specifying the access right to the resources that the user can access. Without proper authentication service one cannot assure proper authorization. Password authentication is ineffective against

- Replay attack - Invader copies the stream of communications in-between two parties and reproduces the same to one or more parties.
- Stolen verifier attack - Stolen verifier attack occur when the invader snips the Password verifier from the server and makes himself as an legitimate user [6].

III. Apache Sentry

There is an option to secure Hadoop cluster with Apache Sentry. Sentry is a highly modular system to provide fine grained role based authorization to both data and metadata stored on an Apache Hadoop cluster. It provides authorization required to provide precise levels of access to the right users and applications. Sentry's key benefits include store sensitive data in Hadoop, extend Hadoop to more users, create new use cases for Hadoop and comply with regulations. Also its key capabilities of it are Fine-Grained authorization, Role-Based authorization, Multi-Tenant administration, to separate policies for each database/schema and ability to be maintained by separate admins. Sentry have been proposed and launched by Cloudera Company [19, 20].

IV. Fully Homomorphic Encryption

This paper [3] proposes a design of trusted file system for Hadoop. The design uses the latest cryptography—fully homomorphic encryption technology and authentication agent technology. It ensures the reliability and safety from the three levels of hardware, data, users and operations. The homomorphic encryption technology enables the encrypted data to be operable to protect the security of the data and the efficiency of the application. The authentication agent technology offers a variety of access control rules, which are a combination of access control mechanisms, privilege separation and security audit mechanisms, to ensure the safety for the data stored in the Hadoop file system.,

Fully homomorphic encryption allows multiple users to work on encrypted data in an encrypted form with any operation, but yields the same results as if the data had been unlocked. So, it can be used to encrypt the data for users, and then, the encrypted data can be uploaded to HDFS without worrying that data be stolen when transferring on the network to HDFS. After data processing with MapReduce, the result is still encrypted and safely stored on HDFS.

As the authors themselves admits, currently, because of the computational complexity, data increases seriously and other reasons when using fully homomorphic encryption, it has not been put into practical use. With the development of cryptography, maybe there will be a practical fully homomorphic algorithm program in the near future.

V. Authentication using One Time Pad

In paper [6], a novel and a simple authentication model using one time pad algorithm is proposed that removes the communication of passwords between the servers. This model tends to enhance the security in Hadoop environment.

The proposed approach provides authentication service by using one time pad and symmetric cipher cryptographic technique. This approach uses two-server model, with a Registration Server and a Back end Server. The whole process of authentication consists of two parts:

1. Registration Process

2. Authentication Process

During the registration process, the user enters his Username and Password. The Password is encrypted (Cipher Text 1) using one-time pad algorithm. Cipher Text 1 is again encrypted using mod 26 operations (Cipher Text 2) and stored in the Registration Server. Again, encrypt the onetime pad key using the Password which results in (Cipher Text 3) using symmetric cipher technique. Cipher Text 3 will be sent to the Backend Server to be stored along with the Username.

Next during the authentication process, after receiving the Username from the user, the Registration Server sends the Username to the user. The Backend Server sends the corresponding Cipher (Cipher Text 3) to the User via Registration Server. The user deciphers it using his Password and returns the key to Registration Server.

Registration Server decrypts Cipher Text 1 with the key returned by the User. Again encrypts the Password with same key and send the Cipher (Cipher Text 4) to the Backend Server.

The Backend server compares Cipher Text 4 with Cipher Text 3. If it matches, sends the Username to the Registration Server. The Registration Server compares the Username with the Username entered by the user. If it matches, the user is authenticated. The random is valid only for one session. Once the user logs out, a new random key replaces the old one.

VI. Accessing HDFS Based on Attribute-Group

Paper [5] is based on the CP-ABE[6] and HDFS designing a secure cloud storage the cipher text control program. On the basis of the CP-ABE and symmetric encryption algorithm (such as AES), they had proposed a cloud-oriented storage efficient dynamic access control scheme cipher text.

In this paper, the properties of the cipher text CP-ABE encryption algorithm based on cloud storage data security access control scheme. Compared to the data owner directly distributed key distribution, centralized management of key distribution method and NameNode-based CP-ABE, easier to manage keys, but also more transparent to the user, that allows users to less involved key generation, key distribution, and other matters. There is certain credibility, requiring CSP must be faithful to run the program and visits Asked the agreement, yet may spy the contents of the data file, and assumes that all parameters and between the communication channel is secure.

Cloud storage security issues affecting the development of cloud storage, reasonable and effective data security access control method can improve the trust of the users for cloud storage services. To solve the security issues of network features and data sharing features in cloud storage service, this paper proposes a data security access scheme storage Based on Attribute-Group in cloud, so that the data owners do not participate in the specific operation of the property and user rights, while the re-encrypted transfer to the NameNode-side, reduce the amount of computation and management costs of the client, ensure the confidentiality of user data, and also Achieve its purpose that the cipher text file sharing. Although the attribute-group-based scheme proposed in this paper has high security and reliability, but the efficiency of the implementation has yet to be improved.

VII. Triple Encryption Scheme for Hadoop-based Data

Cloud computing has been flourishing in past years because of its ability to provide users with on-demand, flexible, reliable, and low-cost services. With more and more cloud applications being available, data security protection becomes an important issue to the cloud. In order to ensure data security in cloud data storage, a novel triple encryption scheme is proposed in this paper, which combines HDFS files encryption using DEA and the data key encryption with RSA, and then encrypts the user's RSA private key using IDEA [11].

A novel triple encryption scheme is proposed and implemented, which combines HDFS files encryption using DEA (Data Encryption Algorithm) and the data key encryption with RSA, and then encrypts the user's RSA private key using IDEA (International Data Encryption Algorithm).

In the triple encryption scheme, HDFS files are encrypted by using the hybrid encryption based on DES and RSA, and the user's RSA private key is encrypted using IDEA. The triple encryption scheme is implemented and integrated in Hadoop-based cloud data storage

Principle of Data Hybrid Encryption is that HDFS files are encrypted using a hybrid encryption method, a HDFS file is symmetrically encrypted by a unique key k and the key k is then asymmetrically encrypted by owner's public key. Symmetrical encryption is safer and more expensive than asymmetrical encryption. Hybrid encryption is a compromising choice against the two forms of encryption above. Hybrid encryption uses DES algorithm to encrypt files and get the Data key, and then uses RSA algorithm to encrypt the Data key. User keeps the private key in order to decrypt the Data key.

They have planned to achieve the parallel processing of the encryption and decryption using MapReduce, in order to improve the performance of data encryption and decryption.

VIII. Security Framework in G-Hadoop

G-Hadoop [21] is an extension of the Hadoop MapReduce framework with the functionality of allowing

[6] Ciphertext-Policy Attribute-Based Encryption

the MapReduce tasks to run on multiple clusters in a Grid environment. However, G-Hadoop simply reuses the user authentication and job submission mechanism of Hadoop, which is designed for a single cluster and hence does not suit for the Grid environment.

The G-Hadoop prototype currently uses the Secure Shell (SSH) protocol to establish a secure connection between a user and the target cluster. This approach requires a single connection to each participating cluster, and users have to log on to each cluster for being authenticated. Therefore, we designed a novel security framework for G-Hadoop.

The paper [22] proposes a new security model for G-Hadoop. The security model is based on several security solutions such as public key cryptography and the SSL (Secure Sockets Layer) protocol, or the concepts of other security solutions such as GSI (Globus Security Infrastructure). Some concepts, for example, proxy credentials, user session, and user instance, are applied in this security framework as well to provide the functionalities of the framework and is dedicatedly designed for distributed environments like the Grid. This security framework simplifies the user authentication and job submission process of the current G-Hadoop implementation with a single-sign-on approach. In addition, the designed security framework provides a number of different security mechanisms to protect the G-Hadoop system from traditional attacks as well as abusing and misusing. Figure 1 shows its security architecture.

Our security model follows the authentication concept of the Globus Security Infrastructure (GSI), while using SSL for the communication between the master node and the CA (Certification Authority) server. GSI is a standard for Gird security. It provides a single sign-on process and an authenticated communication by using asymmetric cryptography as the base for its functionality. As a security standard, GSI adopts several techniques to meet different security requirements. This includes the authentication mechanisms for all entities in a Grid, integrity of the messages that are sent within a Grid, and delegation of the authority from an entity to another. A certificate, which contains the identity of users or services, is the key of the GSI authentication approach. The user needs only provide his user name and password or simply log on to the master node; jobs can then be submitted to the clusters without requesting any other resources. A secure connection between the master node and slave nodes is established by the security framework using a mechanism that imitates the SSL handshaking phase.

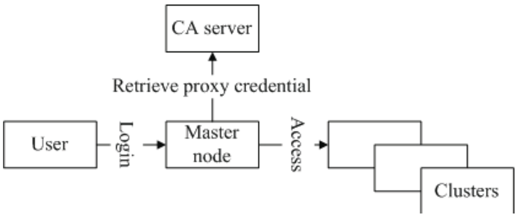

Figure 1. The G-Hadoop security architecture [22]

IX. SUMMARY & CONCLUSION

In this paper, we reviewed security of Apache Hadoop platform including its present security situation, threats and some methods enhancing its security level.

Some challenges in designing security mechanism for Hadoop and improving its security seems to be the following factors [6]:

i. Scale of the system is large.

ii. Hadoop is a distributed file system, so file is partitioned and distributed through the cluster.

iii. Next job execution may be done on a different node from which the user has been authenticated and the job has been submitted.

iv. Tasks from different users may be executed on a single node.

v. Users can access the system through some workflow system.

Altogether, Securing a Hadoop cluster starts with identifying what type data (PII, security sensitive, web blogs or low value-high volume data) will be stored in a Hadoop cluster. Then considering how users will access the data, for example through a middleware application or directly, What are the controls placed in the middleware and whether these controls are sufficient and then deciding about choosing one or some or all of security approaches described above. If controls aren’t sufficient, or the consequences of a breach are high, enabling Kerberos and putting a firewall around Hadoop cluster can be useful. Since often an end-user does not have a line of sight to an enterprise DB, a Hadoop cluster may need to be secured similarly. Then having applications accessing Hadoop services over REST, Apache Knox can be very appreciate to put it between the application and the Hadoop cluster. Currently there are authorization controls at various layers in Hadoop, From ACL in MR to HDFS permission, and more access controls improvements are coming. Enabling wire encryption to protect data as it moves in Hadoop or using custom and other solutions for encrypting data at rest (as it sits in HDFS) can be considered [23].

At last, we can claim that Hadoop has strong security at the file system level, but it lacks the granular support needed to completely secure access to data by users and Business Intelligence applications. This problem forces organizations in industries for which security is paramount (such as financial services, healthcare, and government) to choose either leave data unprotected or lock out users entirely. Mostly, the preferred choice is the latter, severely inhibiting access to data in Hadoop [19]. Although to solve the problem of secure access to data, Apache Sentry has been newly proposed and it promises to be successful, due to overcome these difficulties and make Hadoop secure for enterprises, actually new methods are needed to be proposed.

ACKNOWLEDGMENT

The authors would like to thank all those who contributed to this paper. Further to this, we gratefully acknowledge those in the Cloud Research Center at the Department of Computer engineering and Information Technology, Amirkabir University, IRAN and Cloud Computing lab in University of Tabriz, IRAN.

REFERENCES

[1] P. Mell and T. Grance, "Draft NIST working definition of cloud computing," *Referenced on June. 3rd,* vol. 15, 2009.

[2] M. J. Carey, "Declarative Data Services: This Is Your Data on SOA," in *SOCA*, 2007, p. 4.

[3] S. Jin, S. Yang, X. Zhu, and H. Yin, "Design of a Trusted File System Based on Hadoop," in *Trustworthy Computing and Services*, ed: Springer, 2013, pp. 673-680.

[4] *Apache™ Hadoop®!* Available: http://hadoop.apache.org/

[5] H. Zhou and Q. Wen, "Data Security Accessing for HDFS Based on Attribute-Group in Cloud Computing," in *International Conference on Logistics Engineering, Management and Computer Science (LEMCS 2014)*, 2014.

[6] N. Somu, A. Gangaa, and V. S. Sriram, "Authentication Service in Hadoop Using one Time Pad," *Indian Journal of Science and Technology,* vol. 7, pp. 56-62, 2014.

[7] E. B. Fernandez, "Security in data intensive computing systems," in *Handbook of Data Intensive Computing*, ed: Springer, 2011, pp. 447-466.

[8] M. R. Jam, L. M. Khanli, M. K. Akbari, E. Hormozi, and M. S. Javan, "Survey on improved Autoscaling in Hadoop into cloud environments," in *Information and Knowledge Technology (IKT), 2013 5th Conference on*, 2013, pp. 19-23.

[9] M. Yuan, "Study of Security Mechanism based on Hadoop," *Information Security and Communications Privacy,* vol. 6, p. 042, 2012.

[10] B. Noland. (2013). *6 ways to exploit Hive and what to do about*. Available: http://www.slideshare.net/cloudera/deploying-enterprisegrade-security-for-hadoop

[11] C. Yang, W. Lin, and M. Liu, "A Novel Triple Encryption Scheme for Hadoop-Based Cloud Data Security," in *Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on*, 2013, pp. 437-442.

[12] (2014). *Securing your Hadoop Infrastructure with Apache Knox*. Available: http://hortonworks.com/hadoop-tutorial/securing-hadoop-infrastructure-apache-knox/

[13] V. Shukla, "Hadoop Security Today & Tomorrow," ed: Hortonworks Inc., 2014.

[14] (2014-06-13). *Knox Gateway*. Available: http://knox.apache.org/

[15] X. Zhang, "Secure Your Hadoop Cluster With Apache Sentry," ed: Cloudera, April 07, 2014.

[16] (2014). *Comprehensive and Coordinated Security for Enterprise Hadoop*. Available: http://hortonworks.com/labs/security/

[17] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans*, et al.*, "Apache hadoop yarn: Yet another resource negotiator," in *Proceedings of the 4th annual Symposium on Cloud Computing*, 2013, p. 5.

[18] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM,* vol. 51, pp. 107-113, 2008.

[19] S. V. a. B. Noland. (July 24, 2013). *With Sentry, Cloudera Fills Hadoop's Enterprise Security Gap*. Available: http://blog.cloudera.com/blog/2013/07/with-sentry-cloudera-fills-hadoops-enterprise-security-gap/

[20] (2014). *Security for Hadoop*. Available: http://www.cloudera.com/content/cloudera/en/solutions/enterprise-solutions/security-for-hadoop.html

[21] L. Wang, J. Tao, H. Marten, A. Streit, S. U. Khan, J. Kolodziej, *et al.*, "MapReduce across distributed clusters for data-intensive applications," in *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*, 2012, pp. 2004-2011.

[22] J. Zhao, L. Wang, J. Tao, J. Chen, W. Sun, R. Ranjan*, et al.*, "A security framework in G-Hadoop for big data computing across distributed Cloud data centres," *Journal of Computer and System Sciences,* vol. 80, pp. 994-1007, 2014.

[23] V. Shukla. (Feb. 2014). *Hadoop Security : Kerberos or Knox or both*. Available: http://hortonworks.com/community/forums/topic/hadoop-security-kerberos-or-knox-or-both/