ISSN:0975-9646

Securing Big Data Hadoop: A Review of Security Issues, Threats and Solution

Priya P. Sharma

Information Technology Department
SGGS IE&T, Nanded, India

Chandrakant P. Navdeti

Information Technology Department
SGGS IE&T, Nanded, India

Abstract— Hadoop projects treat Security as a top agenda item which in turn represents which is again classified as a critical item. Be it financial applications that are deemed sensitive, to healthcare initiatives, Hadoop is traversing new territories which demand security-subtle environments. With the growing acceptance of Hadoop, there is an increasing trend to incorporate more and more enterprise security features. In due course of time, we have seen Hadoop gradually develop to label important issues pertaining to, what we summarize as 3ADE (authentication, authorization, auditing, and encryption) within a cluster. There is no dearth of Production environments that support Hadoop Clusters.In this paper, we aim at studying "Big Data" security at the environmental level, along with the probing of built-in protections and the Achilles heel of these systems, and also embarking on a journey to assess a few issues that we are dealing with today in procuring contemporary Big Data and proceeds to propose security solutions and commercially accessible techniques to address the same.

Keywords**—Big Data, SASL, delegation, sniffing, cell level, variety, unauthorized**

I. INTRODUCTION

So, what exactly is "Big data". Put in simple words, it is described as mammoth volumes of data which might be both structured and unstructured. Generally, it is so gigantic that is provides a challenge to process using conventional database and software techniques. As witnessed in enterprise scenarios, three observations can be inferred;

1. The data is stupendous in terms of volumes.
2. It moves at a very fast pace.
3. It outpaces the prevailing capacity.

The volumes of Big Data are on a roll, which can be inferred from the fact that as far back in the year 2012, there were a few dozen terabytes of data in a single dataset, which has interestingly been catapulted to many petabytes today.

To carter to the demands of the industry, new manifestos of manipulating "Big Data" are being commissioned.

Quick fact: 5 exabytes (1 Exabyte = $1.1529*10^{18}$ bytes) of data were created by humans until 2003. Today this amount of information is created in two days [8, 16]. In 2012, digital world of data was expanded to 2.72 zettabytes (1021 bytes). It is predicted to double every two years, reaching the number about 8 zettabytes of data by 2015 [8, 16]. With an increase in the data, there is a corresponding increase in the applications and framework to administer it. This gives rise to new vulnerabilities that need being responded to.

Not only Big Data is about the size of data but also includes data variety and data velocity. Together, these three attributes form the three V's of Big Data

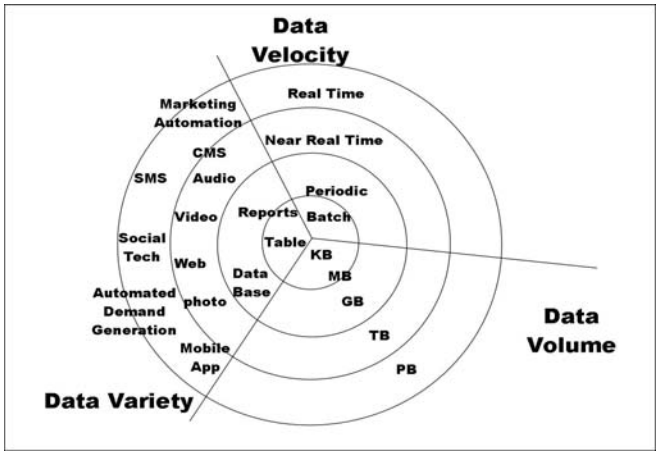

Fig.1 Three V's Of Big-Data [17]

Each of the V's represented in Figure 1 are depicted as below:

Volume or the size of data in present time is larger than terabytes and petabytes. That data is comes from machines, networks and human interaction on systems like social media the volume of data to be analysed is very huge. [8]

Velocity defines the speed of data processing, is required not only for big data, but also all processes, and involves, real time processing,batch processing.

Variety refers todifferent types of data from different or many sources both structured and unstructured. In Past data was stored from sources like spreadsheets and databases. Now in this data comes in the form of emails, pictures, audio, videos, monitoring devices, PDFs, etc. This multifariousness of unstructured data creates problems for storage, mining and analysing the data. [8]To process the large volume of data from different sources, for fast processing Hadoop is used.

Hadoop is a free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment. Hadoopallows running applications on systems with thousands of nodes with thousands of terabytes of data [2]. Its distributed file system supports fast data transfer rates among nodes and allows the system to continue operating uninterrupted at times of node failure.

Hadoop consists of distributed file system, data storage and analytics platforms and a layer that handles parallel computation, rate of flow (workflow) and configuration administration [8]. HDFS runs across the nodes in a Hadoop cluster and together connects the file systems on many input and output data nodes to make them into one big file system [2]. The present Hadoop ecosystem (as shown in fig 2.) consists of the Hadoop kernel, Map-Reduce, the Hadoop distributed file system (HDFS) and a number of related components such as Apache Hive, HBase, Oozie, Pig and Zookeeper and these components are explained as below[7,8]:

- *HDFS*: A highly faults tolerant distributed file system that is responsible for storing data on the clusters.
- *MapReduce:* A powerful parallel programming technique for distributed processing of vast amount of dataon clusters.
- *HBase*: A column oriented distributed NoSQL database for random read/write access.
- *Pig*: A high level data programming language for analyzing data of Hadoop computation.
- *Hive*: A data warehousing application that provides a SQL like access and relational model.
- *Sqoop*: A project for transferring/importing data between relational databases and Hadoop.
- *Oozie*: An orchestration and workflow management for dependent Hadoop jobs.
-

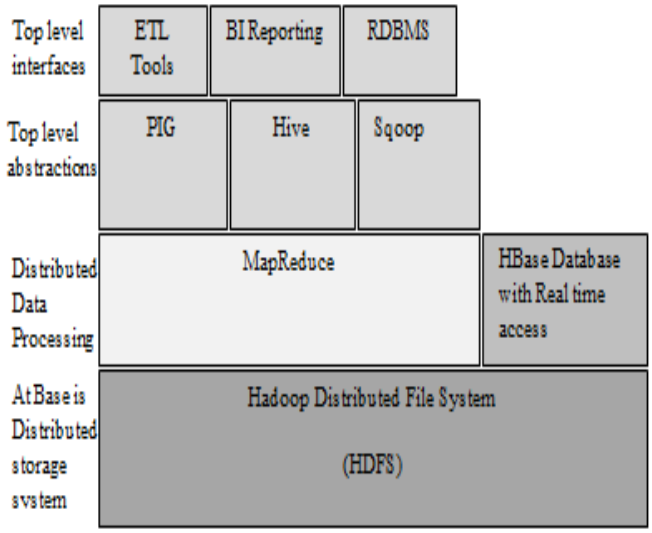

Fig. 2 Hadoop Architecture

The paper is organised asfollows: In section II we describe Big Data Hadoop traditional security and also discuss weakness of the same, security threats, we have describe various security issues in Section III, Section IV we present our analysis of security solution for each of the hadoop components in tabular format and section V is also an analysis of security technologies used to secure Hadoop. Finally we conclude in section VI.

II. BIG DATA HADOOP 'S TRADITIONAL SECURITY

A. *Hadoop Security Overview*

Originally Hadoop was developed without security in mind, no security model, no authentication of users and services and no data privacy, so anybody could submit arbitrary code to be executed. Although auditing and authorization controls (HDFS file permissions and ACLs) were used in earlier distributions, such access control was easily evaded because any user could impersonate any other user. Because impersonation was frequent and done by most users, the security controls measures that did subsist were not very effective. Later authorization and authentication was added, but that to have some weakness in it. Because there were very few security control measures within Hadoop ecosystem, many fortuity and security incidents happened in such environments. Well-meant users can make mistakes (e.g. deleting massive amounts of data within seconds with a distributed delete). All users and programmers had the same level of access privileges to all the data in the cluster, any job could access any of the data in the cluster, and any user could read any data set [4]. Because MapReduce had no concept of authentication or authorization, an impish user could lower the priorities of other Hadoop jobs in order to make his job complete faster or to be executed first – or worse, he could kill the other jobs.

Hadoop is an entire eco-system of applications that involves Hive, HBase, Zookeeper, Oozie, and Job Tracker, and not just a single technology. Each of these applications requires hardening. To add security potentials or capabilities into a big data environment, functions need to scale with the data. Supplementary security does not scale well, and simply cannot keep up. [6]

The Hadoop community supports some security features through the current Kerberosimplementation, the use of firewalls, and basic HDFS permissions and ACLs [5]. Kerberos is not a compulsory requirement for a Hadoop cluster, making it possible to run entire clusters without deploying or implementingany security. Kerberos is also not very easy to install and configure on the cluster, and to integrate with Active Directory (AD) and Lightweight Directory Access Protocol, (LDAP) services. [6]

This makes security problematic to be implemented, and thus limits the adoption of even the most basic security functions forusers of Hadoop. Hadoop security is not properly addressed by firewalls, once a firewall is breached; the cluster is wide-open for attack. Firewalls offer no protection for data at-rest or in-motion within the cluster. Firewalls also offer no protection from security failure which originates from within the firewall perimeter [6]. An attacker who can enter the data centre either physically or electronically can steal the data they want, since the data is un-encrypted and there is no authentication enforced for access[6, 10].

B. *Security Threats*

We have identified three categories of security violation: unauthorized release of information, unauthorizedmodification of information and denial of

resources. The following are the related areas of threat we identify in Hadoop [7]:

- An unauthorized user may access an HDFS file via the RPC or via HTTP protocols and could execute arbitrary code or carry out further attacks
- An unauthorized client may read/write a data block of a file at a DataNode via the pipeline streaming Data-transfer protocol.
- An unauthorized client may gain access privileges and may submit a job to a queueor delete or change priority of the job.
- An unauthorized user may access intermediatedata of Map job via its task trackers HTTP shuffleprotocol.
- A task in execution may use the host OS interfaces to access other tasks, or would accesslocal data which include intermediate Map outputor the local storage of the DataNode that runs on the same physical node.
- An unauthorized user may eavesdrop/sniff to data packets being sent by Data nodes to client.
- A task or node may masquerade as a Hadoop service component such as a DataNode, NameNode, job tracker, task tracker etc.
- A user may submit a workflow to Oozie as another user.
- DataNodes imposed no access control, a unauthorized user could read arbitrary data blocks from DataNodes, bypassing access control mechanism/restrictions, or writing garbage data to DataNode.[10]

III. SECURITY ISSUES

Hadoop present some unique set of security issues for data centre managers and security professionals. The security issues are depicted below [5, 6]:

1) Fragmented Data: Big Data clusters contain data that portray the quality of fluidity, allowing multiple copies moving to-and-fro various nodes ensuring redundancy and resiliency.The data is available for fragmentation and can be shared across multiple servers. As a result, more complexity is added as a result of the fragmentation which poses a security issue due to the absence of a security model.

2) Distributed Computing: Since, the availability of resources leads to virtual processing of data at any instant or instance where it is available, this progresses to large levels of parallel computation. As a result, complicated environments are created that are at high risks of attacks than their counterparts of repositories that are centrally managed and monolithic, which enables easier security implications.

*3) Controlling Data Access:*Commissioned data environments provision access at the schema level, devoid of finer granularity in addressing proposed users in terms of roles and access related scenarios. Many of the available database security schemas provide role based access.

4) Node-to-node communication: A concern with Hadoop and a variety of players available in this field is that, they don't implement secure communication; they bring into use the RPC (Remote Procedure Call) over TCP/IP.

5) Client Interaction: Communication of client takes place with resource manager, data nodes. However, there is a catch. Even though efficient communication is facilitated by this model, it makes cumbersome to shield nodes from clients and vice-versa and also name servers from nodes. Clients that have been compromised tend to propagate malicious data or links to either service.

6) Virtually no security:Big data stacks were designed with little or no security in mind. Prevailing big data installations are built on the web services model, with few or no facilities for preventing common web threats making it highly susceptible.

IV. HADOOP SECURITY SOLUTION

Hadoop is a distributed system which allows us to store huge amounts of data and processing the data in parallel. Hadoop is used as a multi-tenant service and stores sensitive data such as personally identifiable information or financial data. Other organizations, including financial organizations, using Hadoop are beginning to store sensitive data on Hadoop clusters. As a result, strong authentication and authorization is necessary. [7]

The Hadoop ecosystem consists of various components. We need to secure all the other Hadoop ecosystem components. In this section, we will look at the each of the ecosystem components security and the security solution for each of these components, each component has its own security challenges, issues and needs to be configured properly based on its architecture to secure them. Each of these hadoop components has end users directly accessing the component or a backend service accessing the Hadoop core components (HDFS and Map-Reduce).

We have done a security analysis of hadoop components and a brief study of built-in security of the Hadoop ecosystem and we see that hadoop security is not very strong, so in this paper we provide with a security solution around the four security pillars i.e. authentication, authorization, encryption and audits (we summarize as 3ADE), for each of the ecosystem components. This section describes the four pillars (sufficient and necessary) to help secure the Hadoop cluster, we will narrow our focus and take a deep dive into the built-in and our proposed security solution for the Hadoop ecosystem

A. Authentication

Authentication is verifying user or system identity accessing the system. Hadoop provides Kerberos as a primary authentication. Initially SASL/GSSAPI was used to implement Kerberos and mutually authenticate users, their applications, and Hadoop services over the RPC connections [7]. Hadoop also supports "Pluggable" Authentication for HTTP Web Consoles meaning that implementers of web applications and web consoles could implement their own authentication mechanism for HTTP connections. This includes but was not limited to HTTP SPNEGO authentication. The Hadoop components support

SASL Framework i.e. the RPC layer can be changed to support the SASL based mutual authentication viz. SASL Digest-MD5 authentication or SASL GSSAPI/Kerberos authentication.

MapReducesupports Kerberos authentication, SASL Digest MD-5 authentication, and also Delegation token authentication on RPC connections. In HDFS communications between the NameNode and DataNodes is over RPC connection and mutual Kerberos authenticationis performed between them [15]. HBase supports SASL Kerberos secure client authentication via RPC, HTTP. Hive supports Kerberos and LDAP authentication for the user authentication and authentication via Apache Knox explained in section V.

Pig uses the user credentials to submit the job to Hadoop. So there is no need of any additional Kerberos security authentication required but before starting Pig the user should authenticate with KDC and get a valid Kerberos ticket [15]. Oozie provides user authentication to the Oozie web services. It also provides Kerberos HTTP Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO) authentication for web clients. SPNEGO protocol is used when a client application wants to authenticate to a remote server, but is not sure of the authentication protocols to use. Zookeeper supports SASL Kerberos authentication on RPC connections. Hue offers SPENGO authentication,LDAP authentication, it now also supports SAML SSO authentication[15].

There are a number of data flows involved in Hadoop authentication – Kerberos RPC authentication mechanism is used for users authentication, applications and Hadoop Services, HTTP SPNEGO authentication is used for web consoles, and the use of delegation tokens [10]. Delegation token is a two party authentication protocol used between user and NameNode for authenticating users, it is simple and more effective than three party protocol used by Kerberos [7, 15].Oozie and HDFS,MapReduce supports delegation token.

B. Authorization and ACLs

Authorization is a process of specifying access control privileges for user or system. In Hadoop, access controls is implemented by using file-based permissions that follow the UNIX permissions model. Access control to files in HDFS could be enforced by the NameNode based on file permissions and ACLs of users and groups. MapReduce provides ACLs for job queues; that define which users or groups can submit jobs to a queue and change queue properties. Hadoop offers fine-grained authorization using file permissions in HDFS and resource level access control using ACLs for MapReduce and coarser grained access control at a service level[13].HBase offers user authorization on tables, column families. The user authorization is implemented using coprocessors. Coprocessors are like database triggers in HBase [15]. They intercept any request to the table before and after, now we can use the Project Rhino [V] to extend HBase support for cell level ACLs. In Hive, authorization is implemented using Apache Sentry [V].Pig provides authorization using ACLs for job queues; Zookeeper also offers authorization using node ACLs.Hue provides access control via file system permission; it also offers ACLs for job queue.

Although Hadoop can be set up to perform access control via user and group permissions and Access Control Lists (ACLs), this may not be sufficient for every organization. Now-a-days many organizations use flexible and dynamic access control policies based on XACML and Attribute-Based Access Control [10, 13].Hadoop can now be configured to support RBAC, ABAC access control using some third party (as discussed in this section and section V) framework or tool some of which are discussed in section V. Some of the Hadoop‘s components like HDFS can offer ABAC using Apache Knox and also Hive can support role based access control using Apache Sentry. Zettaset Orchestration a product by Zettaset provides role based access control support and enables Kerberos to be seamlessly integrated into hadoop ecosystem. [6, 15]

TABLE I: ANALYSIS OF SECURITY SOLUTION

	Map-Reduce	**HDFS**	**HBase**	**Hive**	**Pig**	**Oozie**	**Zookeeper**	**Hue**
Authentication	MD5-Digest, GSSAPI (Kerberos), Delegation tokens	SASL framework , Delegation tokens	Kerberos, SASL (secure client authentication)	Apache Knox, LDAP authenticat ion	User level permissions	Delegation tokens, Kerberos	Kerberos authenticatio n at RPC layer	Kerberos (Pluggable)
Authorization	Job & Queue ACL (resource level)	POSIX permissions, ABAC	HBase ACLs on tables, columns, families	Apache sentry	ACLs, Apache Sentry	ACLs and FS permissions	ACLs	ACLs and FS permissions
Encryption of data at rest	---	AES, OS level	Third party solution	Third party solution	Third party solution	Third party solution	N/A	Third party solution
Encryption of data in transit	RPC – SASL, HTTPS	RPC – SASL, Data transfer protocol	SASL (secure RPC)	Third party solution	SASL	SSL/TLS	Third party solution	HTTPS
Audit Trails	Yes (Base audit)	Yes (Base audit)	No (But Third party solution can be used)	Yes (Hive metastore)	Third party solution	Yes (services)	Third party solution	Yes(Hue logs)

C. Encryption

Encryption ensures confidentiality and privacy of user information, and it secures the sensitive data in Hadoop. Hadoop is a distributed system running on distinct machines, which means that data must be transmitted over the network on a regular basis, there is an increasing need of demand to move sensitive information into the Hadoop ecosystem to generate valuable perceptions. Sensitive data within the cluster needs special kind of protection and should be secured both at rest and in motion [10]. This data needs to be protected during the transfer to and from the Hadoop system. The simple authentication and security layer (SASL) authentication framework is used for encrypting the data in motion in hadoop ecosystem. SASL security gives guarantee of the data being exchanged between client and servers and make sure that, the data is not readable by a "man-in-middle".SASL supports various authentication mechanisms, for example, DIGEST-MD5, CRAM-MD5, etc. The data at rest can be protected in two ways: First, when file is stored in Hadoop, the complete file can be encrypted first and then stored in Hadoop. In this approach, the data blocks in each DataNode can't bedecrypted until we put all the blocks back and create the entire encrypted file. Second, to applying encryption to data blocks once they are loaded in Hadoop system [15].

Hadoop supports encryption capability for various channels like RPC, HTTP, and Data Transfer Protocol for data in motion. Hadoop Crypto Codec framework and Crypto Codec Implementation have been introduced to support data at rest encryption. HDFS supports AES, OS level encryption for data at rest.Zookeeper, Oozie, Hive, HBase, Pig, Huedon't offer data at rest encryption solution but for this components encryption can be implemented via custom encryption techniques or third party tools like Gazzang's zNcryptor using crypto codec framework's. File system level security, and encryption and decryption of files can be performed on the fly using eCryptfs and Gazzang's zNcrypt tools which are commercial security solution available for Hadoop clusters [10, 13, 15].

To protect data in transit and at rest, encryption and masking techniques can be implemented. Tools such as IBM Optim and Dataguise provide data maskingfor enterprise data [15]. Intel's distribution offers encryption and compression of files [15]. Project Rhino enables block-level encryption similar to Dataguiseand Gazzang. [5]

D. Audit Trails

Hadoop cluster hosts sensitive information, security of this information is utmost important for organizations to have a successful secure big data journey. There is always a possibility of occurrence of security breaches by unintended, unauthorized access or inappropriate access by privileged users. [13] So to meet the security compliance requirements, we need to audit the entire Hadoop ecosystem on a periodic basis and deploy or implement a system that does log monitoring.

HDFS and MapReduce provide base audit support. Apache Hive metastore maintains audit (who/when) information for Hive interactions [13, 15]. Apache Oozie, the workflow engine, provides audit trail for services, workflow submission is maintained into Oozie log files. Hue also supports audit logs.For those Hadoop components which donot provide built-in audit logging, we can use audit logs monitoring tools. Scribe and LogStash are open source tools that integrate into most big data environments, as numbers of commercial products do. So one just need to need to find a compatible tool, get it install, integrate it with other systems like log management, and then actually review the results,and what could went wrong. Cloudera Navigator by Cloudera is popular commercial tool that provides audit logging for big data environment. Zettaset orchestration provides centralized configuration management, logging, and auditing support. [6][15]

V. SECURITY TECHNOLOGIES -SOLUTION FOR SECURING HADOOP

In this we will look at overview of the various commercial and open source technologies that are available to address the various security aspects of big data Hadoop [15].

A. Apache Sentry

Apache sentry an open source project by Cloudera is an authorization module for Hadoop that offers the granular, role-based authorization required to provide precise levels of access to the right users and applications. It support for role-based authorization, fine-grained authorization, and multi-tenant administration [11][15].

B. Apache Knox

The Apache Knox Gateway is a system that provides a single point of authentication and access for various Hadoop services in a cluster. It provides a perimeter security solution for Hadoop. The second advantage is it supports various authentication and token verification scenarios. It manages security across multiple clusters and versions of Hadoop. It also provides SSO solutions, and allows integrating other identity management solutions such as LDAP,Active Directory (AD), and SAMLbased SSO and other SSO systems [9].

C. Project Rhino

Project Rhino provides an integrated end-to-end data security solution to the Hadoop ecosystem. It provides a token based authentication and SSO solution. It offers Hadoop crypto codec framework and crypto codec implementation to provide block level encryption for the data stored in Hadoop. It supports key distribution and management so that MR can decrypt data block and execute the program as per requirement. It also enhances the security of HBase by offering cell level authentication and transparent encryption for table stored in Hadoop. It supports audit logging framework for easy audit trails. [15]

VI. CONCLUSION

In Big Data Era, where data is accumulated from various sources, security is a major concern (critical requirement) as there is no fixed source of data. With the Hadoop gaining larger acceptance within the industry, a natural concern over the security has spread. A growing need to accept and assimilate these security solution and commercial security features has surfaced. In this paper we have tried to cover all the security solutionto secure the Hadoop ecosystem.

REFERENCES

[1] Cloud Security Alliance "Top Ten big Data Security and Privacy Challenges"

[2] Tom White O'Reilly |Yahoo! Press "Hadoop The definitive guide"

[3] Owen O'Malley, Kan Zhang, Sanjay Radia, Ram Marti, and Christopher Harrell "Hadoop Security Design"

[4] Mike Ferguson "Enterprise Information Protection - The Impact of Big Data"

[5] Vormetric "Securing Big Data: Security Recommendations for Hadoop and NoSQL Environments ,October 12, 2012"

[6] Zettaset "The Big Data Security Gap: Protecting the Hadoop Cluster"

[7] Devaraj Das, Owen O'Malley, Sanjay Radia, and Kan Zhang "Adding Security to Apache Hadoop"

[8] Seref SAGIROGLU and Duygu SINANC "Big Data: A Review Collaboration Technologies and Systems (CTS), 2013 International Conference ,May 2013"

[9] Horton works "Technical Preview for Apache Knox Gateway"

[10] Kevin T. Smith "Big Data Security : The Evolution of Hadoop's Security Model"

[11] M. Tim Jones "Hadoop Security and Sentry"

[12] Victor L. Voydock and Stephen T. Kent "Security mechanisms in high-level network protocols. ACM Comput. Surv.1983".

[13] Vinay Shukla s "Hadoop Security: Today and Tomorrow"

[14] MahadevSatyanarayanan "Integrating security in a large distributed system.ACM Trans. Comput. Syst., 1989"

[15] Sudheesh Narayana, Packt Publishing "Securing Hadoop-Implement robust end-to-end security for your Hadoop ecosystem"

[16] S. Singh and N. Singh, "Big Data Analytics", 2012 International Conference on Communication, Information & Computing Technology Mumbai India, IEEE, October 2011

[17] jeffhurtblog.com "three-vs-of-big-data-as-applied-conferences, July 7,2012"