A Survey on Security of the Hadoop Framework in the Environment of Bigdata

Saritha Gattoju1, Dr.NagaLakshmi Vadlamani2
1Research Scholar, Department of Computer Science, GITAM Institute of Science, GITAM (Deemed to be University), Visakhapatnam, India.
2Professor, Department of Computer Science, GITAM Institute of Science, GITAM (Deemed to be University), Visakhapatnam, India.

Email: saritha760@gmail.com

Abstract: The world is becoming increasingly digital at the moment. Every day, a significant amount of data is generated by everyone who uses the internet nowadays. The data are critical for carrying out day-to-day operations, as well as assisting corporate management in achieving their objectives and making the best judgments possible based on the information gathered. Big Data is the process of merging many hardware and software solutions to deal with extremely huge amounts of data that surpass storage capability. It's possible that large amounts of data will be generated. Hadoop systems are used in a variety of areas, including healthcare, finance, and government. insurance, and social media, in order to provide a quick and cost-effective big data solution. The Apache Hadoop is a framework for storing and processing data, managing, and distributing large amounts of information over a large number of server nodes. Here are some solutions that work on top of the Apache Hadoop stack to guarantee data security. To get a complete picture of the problem, we decided to conduct an investigation into existing security solutions for Apache Hadoop security in sensitive data which is stored on a huge data platform employing distributed computing on a cluster of commodity devices. The goal of this paper is to provide knowledge of security and Big Data issues.

Keywords: Yarn, kerberos, HTTPS, HUE, Project Rhino, Security, Encryption.

1. Introduction
Big Data is a massive pool[1] of data that may be a valuable asset to any company. Big data is the process of archiving a large amount of data using the Hadoop framework and a variety of analytics tools that are faster than traditional analytic methods. The five "Vs" of big data are Volume, Velocity, Variety, Varsity, and Value.[2]

Volume: Amount of data: an ever-increasing amount of data from a variety of sources.
Velocity: The pace of data determines the sorts of Big Data processing required, such as The velocity of Big Data is shown through batch processing, real-time processing, and interactive processing.
Variety: Data source heterogeneity allows for a variety of data kinds, including structured, semi-structured, and unstructured data.
Varsity: Big Data's consistency and trustworthiness reveal its genuineness.
Value: Refers to the interaction of new ideas gleaned from various datasets.
2. Hadoop Framework:
Hadoop is an Apache framework that allows enormous data volumes to be processed across clusters of computers using programming concepts.[3] Hadoop is a Java-based open source distributed computing platform with two modules: Map Reduce is used to handle large data volumes, while HDFS is used to store data across distributed clusters of devices. [4] Users can utilise MapReduce to efficiently reduce data. Hadoop has since grown into one of the most current systems for storing, processing, and analysing massive amounts of data using a cluster-based architecture. Large corporations[5], such as eBay (which uses a 532-node Apache Hadoop cluster), Facebook. [6]. The cluster's size expands horizontally and vertically as demand develops. When clusters have been extended, network risks and vulnerabilities have increased (scaled out). Some Hadoop distributions tailored to specific vendors increased performance. The official edition of Apache Hadoop was upgraded by certain vendor-specific Hadoop distributions to fulfil industry expectations for security and centralised administration. [7]

2.1 HDFS:
For storage, HDFS is employed [8,9]. Data processing is handled by Map Reduce, and cluster resource management is handled by YARN. distributed file system hadoop [2.1] (HDFS) It is used to distribute huge files among a group of Data Nodes, each of which has local storage. The Name Node divides the original file into a block with a default size of 64 MB and replicates it to the other Data Nodes based on preset criteria during this procedure.

2.2 MapReduce:
The MapReduce architecture [11] is a master-slave parallel processing system. One master (Job Tracker) and three slaves (Task Tracker) make up a cluster, with one daemon per slave. Any jobs that are disrupted are monitored and rescheduled by the Job Tracker. MapReduce can be used directly for business analytics; for example, transforming data into an optimal format is a popular use case. Optimum format for analytics.

2.3 YARN:
YARN is an Apache Hadoop sub-project that is yet another resource negotiator. In 2012, MapReduce was separated into two parts: YARN and MapReduce [12] The main concept behind YARN is to separate the functions of resource management and job scheduling into separate daemons. The role of a resource manager is to arbitrate sources across all system programmes with the help of a resource manager. To keep an eye on the stream of the CPU, Memory, Utilisation on each node, Disc, and network resource. The resource manager's two most important components are the scheduler and the application manager.

![Hadoop Ecosystem](image)

**Figure.1 Hadoop Ecosystem**
2.4 Zookeeper: Zookeeper is a new tool that helps with the coordination and synchronisation of Hadoop resources or components. The group worked on synchronisation, communication, grouping, and maintenance issues of a network resource, as well as a disc.[12]

2.5 Oozie: Oozie is a tool that assists in the organisation of jobs and unites them as a single entity. Oozie It creates workflows that must be completed in a timely manner. in a systematically organised manner When some data or external stimuli is triggered, coordinator jobs are triggered.[13]

2.6 Pig: Pig is a tool for organising, processing, and analysing huge amounts of data. The Pig Latin language, which was built expressly for this architecture, runs on the JVM. It's built on a query-oriented language called pig Latin, which is similar to SQL.

2.7 Hive: Hive uses SQL technique and an interface to read and write large data sets. Mahout adds Machine Learning capabilities to a system or application. It is particularly scalable since it supports both real-time and batch processing.

2.8 Apache HBase: HBase is a NoSQL database that can handle everything in the Hadoop Database and supports all types of data. It has the capabilities of Google's BigTable, allowing it to operate with large data sets successfully.

2.9 Apache Spark: Many firms employ two tools: Apache Spark and Hadoop. All process-intensive workloads, such as batch processing, interactive or iterative real-time processing, and graph processing, are handled by them. Converters and visualisation It is used interchangeably in most professional environments.

3. Literature

The major purpose of this work is to assure the security of essential data at an HDFS storage level, which has not been done by Kerberos, in today's world of Big Data, where all information is gathered from numerous sources to a single distributed database. It also covered how to safeguard sensitive data and how encryption is used to protect data in the Hadoop cluster from unauthorised access.[9]

This paper focused on cloud data security protection. When storing data in Hadoop, guarantee data security. The suggested method of encryption is used to encrypt HDFS files. The algorithms AES and OTP are utilised. [10]

Using the CP-ABE attribute encryption mechanism and the encryption feature of the ciphertext policy attribute, to create a ciphertext policy attribute-based encryption deployment architecture and propose a group attribute encrypted access control strategy for Hadoop in a cloud environment, and to achieve fine-grained access control for Hadoop in a cloud environment.[12]

Developed a systematic framework for secure exchange of sensitive data on big data platforms, allowing for secure data submission and storage via a heterogeneous proxy re-encryption mechanism, as well as secure use of clear text on the cloud platform via the VMM. The proposed design effectively protects the privacy of users' personal information.[13]

Various techniques to data security in the Hadoop distributed file system are depicted in this article. The first solution uses Kerberos in HDFS to ensure that data blocks are accessed correctly and only by authorised users. Ticket issuing Ticket issuing Ticket issuing Ticket issuing Ticket issuing Ticket is Ticket play an important role in guaranteeing the name node's security. The second strategy, the Bull Eye Algorithm Technique, lays out the security procedure from node to node and monitors the nodes from all angles to prevent attacks. The third option is based on Name node, which ensures security by replicating a name node to prevent future server crashes.[14]

The paper's main focus was on defining the qualities that make a jar file dangerous. Any attempt by a non-root user to read metadata, change file permissions, or perform dfsadmin commands, for example, will be treated as suspicious, and the jars containing such code inside their mappers and reducers will be treated as malicious.

Any attempt by a non-root user to read metadata, change file permissions, or invoke dfsadmin commands, for example, will be deemed suspicious, and the jars containing such code among their mappers and reducers will be regarded malicious.
HDFS security will be strengthened by sandboxing map reduction jobs and other jar files, which will prevent dangerous jars from running.
In our work, a sandboxing feature prevents undesired or harmful jar files from accessing the file system. The shell script can be tweaked to filter jars according to your needs.

Data silos refer to the storage of data or information in separate compartments. These compartments can be found at various levels of management, inside individual departments of organisations, in the government sector, and in a variety of other settings. It will be difficult to find collaboration within different silos; silos are created to meet company goals and to benefit the market. For 98 percent of executives, it is a point of contention.

4. Securing sensitive data in Hadoop:
Sensitive data inside Hadoop can be classified into two high-level categories:

- Sensitive data related to customers’ personal information, customers’ financial information, and so on that exists in enterprise systems and that needs to be brought to Hadoop for analysis.

- The Hadoop analytical process generates sensitive insights after processing the data stored inside Hadoop. These insights are more valuable and sensitive compared to the raw source data that is used to generate them. For example, a retail e-commerce enterprise has detailed transactions of customer purchases. These transaction details might not be very sensitive. This data is brought to Hadoop for generating various insights. Using the customer historical purchases and correlating the same with customer’s household purchases, insights related to customer purchase patterns, behavior patterns, customer sentiment, and customer life events could be inferred. This information is highly sensitive compared to each of the individual transactions. These insights if not secured properly could lead to significant losses for an enterprise.

The insights generated should be classified and encrypted so that only authorized users can access the insights.

- Only a limited set of users who really need to act on sensitive data should have access to these data sets. Data access should be restricted for any non business user (IT user) for these sensitive data sets.

- The intermediate data that is created during the insights generation process should be secured and removed as soon as it is not required.

- Need to track which user has downloaded the sensitive insights and control the life cycle of such downloaded insights. This is required to ensure compliance.

- Access to these sensitive data sets and any other local copy of the insights should be removed once the user ceases to have authorization to access the data sets.

- The data that is in motion during storing and retrieving the data from Hadoop needs to be secured.

5. Securing data at rest:
There are two methods for encrypting Hadoop data sets. Data security at rest stores data by dividing big files into blocks. The files and folders in the local file system can be viewed once the login credentials for the user (root or HDFS) who has access to the data node are compromised. Even if the login credential is compromised, the data saved within the sensitive data sets cannot be accessed. To obtain the full file, the separate blocks can be reassembled.

6. Securing data in motion:
Data is introduced into the Hadoop ecosystem using the Hadoop client, Sqoop, or Flume. This data must be safeguarded throughout its transmission to the Hadoop system. The SASL authentication mechanism is used to encrypt data in transit. SASL is an authentication framework that adds capabilities for authentication to connection-based protocols. SASL security ensures that data sent between clients and servers is encrypted and unreadable by a "man in the middle.”

Typically, an SASL negotiation goes like this:
The client connects to the server and asks for authentication.
The server responds with a list of authentication mechanisms that are supported.
The customer selects one of the authentication methods (for example DIGEST-MD5).
The server will then begin exchanging authentication messages with the client until the authentication is successful or unsuccessful. After successful authentication, the client and server share the session secrets, which are used to encrypt the sent data. For authentication, SSL employs public key cryptography. As a result, both the client and the server must have a shared secret in order to validate the authenticity of the request. SSL secures communication between the client and the server.

Once trust is established between the client and server, SSL enables the encryption of messages transmitted. This ensures that SSL-encrypted communication is secure. To communicate with the Hadoop Name Node, the Hadoop client uses Hadoop RPC (refer to the following figure). SASL security is supported by the Hadoop RPC protocol. This ensures that communication between the Hadoop client and the Name Node is secure and encrypted.

7. Implementing data encryption in Hadoop

Encryption [21] is one of the most important ways for securing sensitive data in Hadoop. Hadoop does not natively enable data encryption; nevertheless, we can use the compression file handling feature to provide support for encryption. We develop a proprietary compression codec that includes encryption. As a result, whenever we need to encrypt data, we must set the compression codec to this specific compression codec. Project Rhino employs a similar technique. A sample compression codec can be found at the following address: https://github.com/delipark/encrypted-hdfs/ by Seonyoung Park and Youngseok Lee. The flow diagram below depicts the approach used to implement custom data encryption in Hadoop. To support bespoke encryption, we expand the Hadoop API's compression codec class and implement encryption within this class.

7.1 Setting Up a Secured Hadoop Cluster

To create a secure [21] Hadoop cluster, we must configure Kerberos authentication on the nodes. Kerberos authentication, which employs the hostname to resolve the principal name, necessitates reverse DNS lookup on all nodes. After installing and configuring Kerberos, we create the Hadoop service principals and all user principals. Following that, we update the Hadoop parameters to allow Kerberos authentication on all nodes and launch the Hadoop cluster.

7.2 Setting up Kerberos

The first step in [21] establishing a secure Hadoop cluster is to configure Kerberos authentication and check that Kerberos authentication for Hadoop service principals works on all nodes in the cluster. To configure Kerberos, we create a Kerberos Server (KDC) on a separate node and install the Kerberos client on all Hadoop cluster nodes, as indicated in the figure:

![Figure.2 Setting up Kerbores](https://github.com/delipark/encrypted-hdfs/ by Seonyoung Park and Youngseok Lee)
7.3 Accessing a secured Hadoop cluster from an enterprise network
The figure below depicts a typical deployment architecture of a secured Hadoop cluster in a business context.

![Diagram of Hadoop Cluster](image_url)

The Hadoop cluster is firewalled with the Corporate Network, and connectivity is only available through the Edge Nodes (also also known as Gateway Servers). The Gateway Server serves as a point of entry for external applications, tools, and users to the secure Hadoop cluster. It's put in place between the Hadoop cluster and the corporate network.[21]

Clients on the corporate network cannot access the Hadoop cluster directly. They connect to the Gateway Servers and perform all Hadoop cluster operations. All client tools, including Hive, Pig, and Oozie, are installed on the Gateway Servers, so users do not need to be granted login access to every node in the cluster, and the cluster nodes are segregated.

7.4 HttpFS
Installing HttpFS on the cluster's Gateway Servers is one approach for providing direct access to Hadoop operations from the corporate network. HttpFS can be used to access data in HDFS on a cluster that is protected by a firewall. HttpFS is a Java-based web application that runs on a preconfigured Tomcat server that comes with the HttpFS binary distribution.[21].
HttpFS works as a proxy and stores data in the cluster using WebHDFS. One of the most significant drawbacks of HttpFS is that we cannot send massive amounts of data using this HTTP interface. When we send data to the cluster, HttpFS runs as a Tomcat-based web application, and the data must pass through this application. To transport big amounts of data, we need use native RPC. HttpFS only allows you to work with the Hadoop Distributed File System (HDFS). When a user needs to run Pig, Hive, or MapReduce on the cluster, they must still log in to the Gateway Server.[21]

7.5 HUE
HUE is the Apache Hadoop UI that is open source. HUE provides a web application that can be used to execute the following tasks:[21]
- Explore the HDFS filesystem; and
- Create workflows that may be uploaded to Oozie.
- Pig editor and executor • Map job browser
- UI for submitting queries to Hive and Impala on the secured Hadoop cluster
- Executor of the Scoop command

One of HUE's drawbacks is its inability to upload and download files from the Hadoop cluster. When installed on the Gateway Server, HUE uses HttpFS internally to proxy the Hadoop cluster. The interaction of end users with HUE is depicted in the diagram below:
7.6 Knox Gateway Server

Knox Gateway [21] is another Apache incubator project that solves the issue of safe access to the Hadoop cluster from corporate networks. Knox Gateway acts as a single point of contact for Apache Hadoop services in a cluster. Knox runs as a server cluster in the DMZ zone, isolating the Hadoop cluster from the rest of the corporate network. Knox Gateway's main feature is that it provides perimeter security for Hadoop REST APIs by restricting the number of network endpoints required to access a Hadoop cluster. As a result, it conceals the internal Hadoop cluster topology from end users. At the perimeter of the network, Knox provides a single point of authentication and token verification.
7.7 Dataguise is a Hadoop data visualisation tool.

Dataguise (DG) is a Hadoop data visualisation tool. Enables data encryption based on symmetric keys. The ability to identify and encrypt sensitive data is one of Data guise’s core features. For sensitive data protection, it provides encryption and masking techniques. It works with Hadoop API, Sqoop, and Flume to encrypt data. As a result, it can be used to encrypt data as it enters and exits the Hadoop ecosystem.

7.8 Gazzang zNcrypt

Gazzang zNcrypt offers transparent block level encryption as well as the ability to manage encryption keys. zNcrypt functions as a virtual filesystem, intercepting any application layer request for file access. It encrypts the block while it is being written to disc. For optimal performance in the cryptographic process, zNcrypt makes use of the Intel AES-NI hardware encryption acceleration. It also supports role-based access control and policy-based encryption key management.

7.9 eCryptfs for Hadoop

The multilayer cryptography filesystem eCryptfs is a Linux filesystem. Cryptographic metadata is stored in the header of each file created by eCryptfs. When encrypted files are copied across hosts, the relevant key from the Linux kernel keyring is used to decrypt them. We can secure a Hadoop cluster by installing eCryptfs on each node. This ensures that data is shared between nodes in a transparent manner, and that all data is encrypted before being written to disc. More information on eCryptfs may be found at the following link: https://launchpad.net/ecryptfs.[15]

8.Project Rhino helps to secure the Hadoop ecosystem.

The goal of Project Rhino was to provide a holistic end-to-end data security view of the Hadoop ecosystem. The following were the essential characteristics:
- Hadoop crypto codec architecture and implementation for block-level encryption of Hadoop data encryption.
- Support for key management and distribution, so MapReduce can decrypt the block and run the programme when it's needed.
- Enhancing HBase security by introducing cell-level authentication and providing transparent encryption for Hadoop HBase tables.
- Audit logging architecture and log formats have been standardised to make audit trail analysis easier.

8.1 Security technologies are mapped to the reference architecture.

We investigated the various commercial [22,23] and open source options for safeguarding the Big Data platform. This section describes how these various technologies are mapped and how they fit into the overall reference design.
9. Infrastructure security

Physical security must be enforced by hand. Unauthorized access to a dispersed cluster, on the other hand, is avoided by establishing Kerberos security in the cluster. Kerberos guarantees that services and users verify their identity with the KDC before gaining access to infrastructure services. Project Rhino intends to expand on this by providing a token-based authentication mechanism.

9.1 OS and file system security

The file encryption technique is used to enforce file system security by creating a secured virtualization layer on top of the current OS filesystem. Those written to the disc are encrypted, and files read from the disc are decrypted in real time. These characteristics are provided by the eCryptfs and zNcrypt tools. SELinux also provides substantial security by hardening the operating system.

9.2 Application security

Sentry and HUE, for example, provide a platform for secure Hadoop access. They integrate with LDAP to enable enterprise-wide integration.

9.3 Network perimeter security

Isolating the Hadoop cluster from the rest of the organisation is a standard strategy for ensuring perimeter security in Hadoop. Users must still connect to the cluster using tools like Knox and HttpFS, which provide a proxy layer for end users to remotely connect to the Hadoop cluster, submit jobs, and access the file system.

9.4 Data masking and encryption

Encryption and masking techniques are used to safeguard data in transit and at rest. Large scale data masking for enterprise data is provided by tools such as IBM Optim and Dataguise. We use block-level encryption in Hadoop to safeguard data in REST. File encryption and compression are supported by Intel's distribution. Project Rhino, like Dataguise and Gazzang, supports block-level encryption.

9.5 Authentication and authorization

While authentication and authorization have advanced greatly, solutions such as Zettaset Orchestrator and Project Rhino enable authentication and authorization interaction with enterprise systems.
9.6 Audit logging, security policies, and procedures

Tools like Cloudera Manager allow common security audit recording for user access to Hadoop Cluster. Cloudera Manager can also produce warnings and events based on the organisational policies that have been set. Similarly, Intel's manager and Zetaset Orchestrate enforce security measures in the cluster in accordance with corporate policies.

9.7 Security Incident and Event Monitoring

It is critical to detect security incidents and monitor occurrences in a Big Data platform. Open source solutions like OSSEC and IBM Gaudium enable a safe Hadoop cluster to identify security issues and integrate easily with enterprise SIEM tools.

10. Conclusion:

Throughout this, we focused on Hadoop security tools, Hadoop security options, and safeguarding the Hadoop environment with Project Rhino. The reference architecture is linked to data encryption, data at rest, data at motion, and security technologies by installing kerberos for securing sensitive data in the Hadoop framework. In this paper, Information and features from big data are used all around the world. In order to present a real-time view of the big data difficulties, the worries are also mentioned. In order to promote awareness, the security issue is being highlighted more. To overcome the challenges created by data aggregation, organisations must devise novel ways to preserve critical tools, processes, and procedures used to gather, maintain, and analyse data.

References:

[1] Nikunj Joshi & Bintu Kadiwal" "A Survey Big Data Security and Privacy Issues" 978-1-5090-5682-8 /17 ©2017 IEEE
[2] Sharif Nawaz Y. Inamdar1, Ajit H. Jadhav2, Robit B. Desai3, Pravin S. Shinde4, Indrajeet M. Ghadage5, Amit A. Gaikwad6 “ Data Security in Hadoop Distributed File System” Volume: 03 Issue: 04 | Apr-2016
[3] Gurjit Singh Bhathal, Amardeep Singh “Big Data: Hadoop framework vulnerabilities, security issues and attacks” 2590-0056/© 2019
[4] Getaneh Berie Tarekgn” Big Data: Security issues, challenges and future scope” Volume 7, Issue 4, July–Aug 2016
[5] Nandhini.P PG Scholar, “A Research on Big Data Analytics Security and Privacy in Cloud, Data Mining, Hadoop and Mapreduce”. Vol. 8, Issue4 (Part -III) April 2018
[6] Mr. Shrikant Rangrao Kadam1 , Vijaykumar Patil.”Review on Big Data Security in Hadoop” Volume: 04 Issue: 01 | Jan -2018
[7] Madhwaraj M Shetty R Manjahia D.H “Data Security in Hadoop Distributed File System” 978-1-5090-3751-3/16 2016 IEEE
[8] Paresh Wankhede Nayanjyoti Paul “Secure and Multi-tenant Hadoop Cluster”2016  (ICGHPC)
[9] Hadeer Mahmoud, Abdelfatah Hegazy Mohamed H. “An approach for Big Data Security based on Hadoop Distributed File system”978-1-5386-0879-1/18 2018 IEEE
[10] Yusuf parvez “The Hadoop Security in Big Data: A Technological Viewpoint and Analysis” Vol.7,Issue.3, pp.1-14, June2019.
[11] Huixiang Zhou Qiaoyan Wen ” A New Solution of Data Security Accessing for Hadoop Based on CP-ABE “ 978-1-4799-3279-5/14 2014 IEEE
[12] Xinhua Dong, Ruixuan Li, Heng He, Wanwan Zhou, Zheng Yuan Xue, and Hao ”Securing Sensitive Data Sharing on a Big Data Platform” Volume 20, February 2015
[13] G. Saritha, Dr. V. Nagalakshmi"Support vector machine and feature selection based optimization framework for big data security"(IJARET) Volume 11, Issue 11. November 2020
[14] B. Saraladevia , N. Pathanirajaa , P. Victor Paula , M.S. Saleem Bashab , P. Dhavachelvan "A Study in Security Perspective Big Data and Hadoop”(ISBCC’15)
[15] Gousiya Begum S. Zahoor Ul Huq and A. P. Siva Kumar”"Sandbox security model for Hadoop file system ”
[16] Seema Rai, Ashok Sharma “International Journal of Engineering and Advanced Technology” (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-3, February, 2020
[17] https://launchpad.net/ecryptfs.
[18] Lingala Thirupathi and Venkata Nageswara Rao Padmanabhuni, “Multi-level Protection (Mip) Policy Implementation using Graph Database” International Journal of Advanced Computer Science and Applications(IJACSA), 12(3), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0120350.
[19] Lingala Thirupathi and P.V. Nageswara Rao, "Developing a MultiLevel Protection Framework Using EDF", International Journal of Advanced Research in Engineering and Technology (IJARET),2020, volume:11, Issue: 10, Pages: 893-902.
[20] Lingala Tirupati, Dr. Venkata Nageswara Rao Padmanabhuni, "Protected framework to detect and mitigate attacks": International journal of analytical and experimental modal analysis, volume XII, Issue VI, (2020) Page No: 2335-2337, DOI: 18.0002.IJAEMA.2020.V12I6.200001.0156858943.

[21] Securing Hadoop Implement robust end-to-end security for your Hadoop ecosystem by Sudheesh Narayanan [PACKT Publishing] open source

[22] Gudipalli Vedavyas, Saritha Gattoju, "Theoretical Foundations of Big Data Analysis" http://www.warse.org/conference/icpet2016/

[23] https://www.congruentsolutions.com/blogposts/25-key-tools-in-apaches