International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-4, February 2019

Security Vulnerabilities in Hadoop Framework

Gousiya Begum, S. Zahoor Ul Huq, A.P. Siva Kumar

Abstract: ***Apache Hadoop emerged as the widely used distributed parallel computing framework for Big Data Processing. Apache Hadoop is an open source framework suitable for processing large scale data sets using clusters of computers. Data is stored in Hadoop using Hadoop Distributed File System. Though Hadoop is widely used for distributed parallel processing of Big Data, some security vulnerabilities does exist. As part of our research we have investigated Hadoop Framework for possible security vulnerabilities and also demonstrated the mechanism to address the identified security vulnerabilities. Our findings include the vulnerabilities in logging mechanism, file system vulnerabilities, and addition of external jar files to the framework. we have addressed these vulnerabilities using custom Map Reduce jobs.***

Index Terms: ***Custom Map Reduce, Hadoop Distributed File System, Hadoop Framework, Security vulnerabilities.***

I. INTRODUCTION

Apache Hadoop is widely used open source distributed parallel processing framework for Big Data processing using clusters of computers. The Core of Apache Hadoop is Map Reduce and it is a distributed massive parallel processing programming model that is used in processing the files stored on Hadoop Distributed File System (HDFS). Map Reduce is java based programming model. A typical architectural view of Hadoop framework is shown in Figure 1.

It is evident from the architecture that data stored in HDFS will be processed by Map Reduce program only. Code written in any other tool like PIG, HIVE etc. will also be converted into a Map Reduce job. Apache Hadoop uses the concept of data locality where the program will be moved to the place where data is available, in contrast with the conventional process of loading data and processing it.

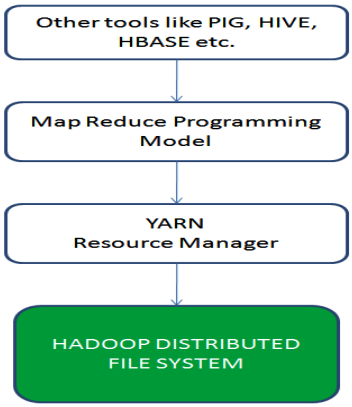

Figure 1: Hadoop Architecture Overview

Since we are dealing with Big Data, moving code is more effective than moving the data for processing. A file is stored in Hadoop using Hadoop Distributed File System (HDFS). In HDFS, a file will be divided into blocks and these blocks are spread across different data nodes in the cluster. Information about the file and file blocks will be maintained in the cluster by Name node using fsimage file. Data nodes will continuously send their heart beat every 3 seconds to share their current status and thus fsimage is always updated with the latest information. Data in HDFS is always accessed as key value pairs. Users of the Hadoop Cluster can access the cluster from their accounts created on one of the node called Edge node. Typical Hadoop cluster is shown in Figure 2.

When user submits a Map Reduce job to the Hadoop Cluster, the job tracker daemon present in the Name node will redirect the job to the task tracker present inside the respective data nodes.

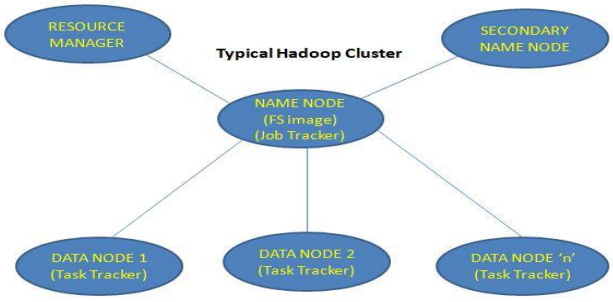

Figure 2. Hadoop Cluster

The instance of Map Reduce job will run separately on each block ensuring distributed massive parallel processing. Any Map Reduce job will contain a Mapper and Reducer where Mapper will process the file blocks and the results generated by the Mappers will be aggregated by the Reducers.

Manuscript published on 28 February 2019.
*Correspondence Author(s)
Gousiya Begum, Research Scholar, JNTUA, Anantapuramu, Assistant Professor, CSE Department, MGIT, Hyderabad, India.
Dr. S. Zahoor Ul Huq**,** Professor, CSE Department, GPREC, Kurnool, India.
Dr. A. P. Siva Kumar, Assistant Professor, CSE Department, JNUTA, Anantapuramu, India.

However the Map Reduce programming model is prone to security vulnerabilities which can lead to tampering of data and data leakages. The security vulnerabilities in the framework are discussed in literature survey and mechanisms to handle the security vulnerabilities are discussed in the implementation section of the paper.

II. LITERATURE SURVEY

We have investigated the Hadoop framework especially HDFS and Map Reduce to identify various security vulnerabilities. Some of the key vulnerabilities were found with respect to four specific parameters viz.

1. Logging mechanism in Map Reduce Program model.
2. Inclusion of external jars through Map Reduce program
3. File System vulnerabilities
4. Password less entry vulnerability

1. Logging mechanism in Map Reduce:

In Hadoop ecosystem, any Big Data processing tool used, inherently converts to a Map Reduce job. It means, only a Map Reduce job will run on the file system. Hence the Map Reduce logs become more important to analyze the effect of Map Reduce job on the data. The existing log mechanism of Map Reduce job will record only the details of the data blocks that gets modified. The present logging mechanism will record the time of the modification, the node from where modifications were triggered. The vulnerability in current logging mechanism is, it will not record the user account from where actual modifications were triggered and no details about the job are stored in the log. This is not enough to identify the actual user who has initiated modifications to the data.

Figure 3: Default log file of YARN(Hadoop 2.0)

The default logging mechanism does not include the complete details of the job and the user account from where the job has been executed. It records only the node from where the job has been triggered. This is a vulnerability of the logging mechanism used in hadoop Map Reduce framework. We have developed customized logging mechanism for Map Reduce job and it's discussed in the implementation section.

2. Inclusion of external jars through Map Reduce program:

As part of execution of Map Reduce job, we may need to load some external jars during runtime. For example to interact with relational data bases like oracle, mysql etc, respective jar files must be loaded during execution of the job. Using a method in org. apache. hadoop. Map Reduce.Job class. The method add Fileto Classpath (Path file) is used to upload any jar to the framework. This method does not screen the jar for presence of any harmful code in the jar files.

3. File System vulnerabilities:

HDFS is one of the robust distributed file systems. However HDFS is also prone to few security vulnerabilities. One of the key vulnerability is the ability to access file system from any user account using file system API. Using file system API we cannot add or modify files on the cluster but the existing files can be read and metadata about the file including file size, date and time of creation, file owner, number of file blocks etc. can be retrieved.

```
Job job = new Job(conf);

job.addFileToClassPath(new Path("/tmp/mysql-connector-java-5.0.8-bin.jar"));
job.setJarByClass(DBacess.class);
job.setMapperClass(Map.class);
job.setReducerClass(Reduce.class);
job.setMapOutputKeyClass(Text.class);
job.setMapOutputValueClass(IntWritable.class);
job.setOutputKeyClass(DBOutputWritable.class);
job.setOutputValueClass(NullWritable.class);
job.setInputFormatClass(DBInputFormat.class);
job.setOutputFormatClass(DBOutputFormat.class);
```

Figure 4: Method to load any jar in to framework

Creating a file system object using file system API

```
URI uri =
URI.create("hdfs://172.16.31.110:8020/datasets");
Configuration conf=new Configuration();
FileSystem fs=FileSystem.get(uri,conf,"st01");
```

Figure 5: File system object creation

The above figure demonstrate the usage of File System API to create a file system object from user account "st01" . Once the file system object is created we can retrieve metadata about any file on the cluster which can lead to serious security and privacy concerns.

4. Password less entry vulnerability

When a fully distributed hadoop cluster is setup, we use a cluster of cooperative computers. As part of the cluster these computers do exchange lot of data. For example data nodes will send their heart beat to Name node every 3 seconds. Name node will load file blocks across various data nodes. In order to provide smooth communication between these computers, password less entry scheme using SSH is employed during the cluster setup.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

SSH key pairs are exchanged by all the computers of the cluster and hence no authentication is done thereafter. This is a security concern since if one machine in the cluster is compromised then it can lead to serious threat to all other machines in the cluster. If the cluster is setup using virtual machines, it can lead to VM Hijacking. VM hijacking refers to the situation where in an unauthorized person gains control over the hypervisor. Hadoop supports Kerberos authentication to authenticate users in accessing any services of the Hadoop cluster. If the attacker succeeds in penetrating in to any one of the data node, then Kerberos is of no use.

III. IMPLEMENTATION

As part of our research we have made an effort to address the security vulnerabilities viz.

1. Logging mechanism in Map Reduce.
2. Inclusion of external jars through Map Reduce program
3. File System vulnerabilities
4. Password less entry vulnerability

1. Logging mechanism in Map Reduce: Conventional logging mechanism does not include the information related to the user account from where a Map Reduce job has been initiated.

To address this problem we have designed a custom Map Reduce logging scheme where we try to record the user name in the log file and hence we can keep track of the modifications and the users responsible for these modifications.

import org.apache.hadoop.MapReduce.Mapper;

import org.apache.log4j.Logger;

public class NewMapper extends Mapper<LongWritable,

Text, Text, Text> {

private static final Logger logger = Logger.get Logger (New

Mapper. class);

protected void Map (Long Writable key, Text value,

Mapper<Long Writable, Text, Text, Text>.Context context)

throws IO Exception, Interrupted Exception {

logger. info (org. apache. hadoop. Map Reduce. Job Context.

Get User());

Now for every Map Reduce job, this block of code is added in the driver class or main method of the driver class to log the user name from where the job has been initiated. This will give a better logging mechanism when compared to the existing logging mechanism. Storage mechanism of log files is also a security vulnerability[1] because these log files are not encrypted and are access able to all users from the YARN portal.

2. Inclusion of external jars through Map Reduce program: *Job.addFilestoClassPath*(Path file) is a method used to load any jar file in to hadoop framework. There is no screening mechanism to understand whether the jar contains any harmful executable codes or not. In order to avoid this vulnerability the one possible method is to deprecate this method and only administrator to add external jars to the cluster. As part of our research work we have identified that jar files created using JDK 1.7 and older versions are prone to security vulnerability due to AtomicReferenceArray Class which can allow any object to be passed. However JDK 1.8 and above versions are not vulnerable to this attack. Hence it is recommended not to use JDK 1.7 and older versions during cluster setup. A simple GUI can be provided for uploading external jars and this GUI must check the java version of the jar and do not allow any jars to be uploaded if they are created using JDK 1.7 or older versions than JDK 1.7. The malicious jars can lead performance degradation of the cluster especially if any of the data node is corrupted [2].

3. File System vulnerabilities:

File system can be easily accessed from any user account using file system API. This can be avoided by restricting access to only the root user and access can be denied for all other users. we can override the get method of file system in our Map Reduce job and ensure that always root user is accessing HDFS from remote machine. We were able to access HDFS file system from user account which does not have privileges of root. The code snippet of the same is shown in Figure 6.

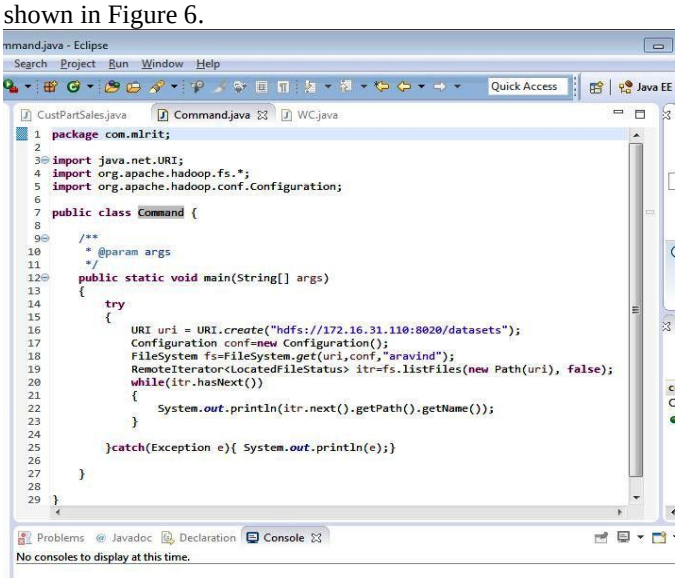

Figure 6: Accessing File system using API

4. Password less entry vulnerability: KERBEROS must be implemented in a more sophisticated manner rather than a simple ticket based delegation model [3]. password less entry may lead to intrusion, disclosure of data blocks and its meta data from Name node. If the meta data is leaked, it will be serious threat to data security and privacy [4]. if at least one block is tampered it can affect the entire file.

IV. CONCLUSION

Some of the key vulnerabilities in hadoop framework were identified and mechanisms to address these vulnerabilities are also discussed. It is very important to go through the API because API play a very significant role in writing customization applications on Map Reduce and HDFS.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Hadoop framework also suffers from insider attacks which is ignored and needs enough attention[5]. Cloudera Distribution for Hadoop (CDH) is the widely used Hadoop distribution and in CDH security is employed using TLS encryption and authentication using Kerberos[6]. CDH security has its own limitations and needs more concrete solutions to address security and privacy issues in Hadoop framework.

REFERENCES

1. Srinivasan, Madhan Kumar, and P. Revathy, "State-of-the-art Big Data Security Taxonomies," *Proceedings of the 11th Innovations in Software Engineering Conference, ACM*, 2018.
2. Wang, Jiayin, et al. "Seina: A stealthy and effective internal attack in hadoop system," *Computing, Networking and Communications (ICNC), 2017 International Conference on. IEEE*, 2017.
3. Parmar, Raj R., et al. "Large-scale encryption in the Hadoop environment: Challenges and solutions," *IEEE Access* 5 (2017): 7156-7163.
4. Rao, P. Ram Mohan, S. Murali Krishna, and AP Siva Kumar. "Privacy preservation techniques in big data analytics: a survey," *Journal of Big Data* 5.1 (2018): 33.
5. Dou, Zuochao, et al. "Robust insider attacks countermeasure for Hadoop: Design and implementation,." *IEEE Systems Journal* 12.2 (2018): 1874-1885.
6. Cloud Security Overview https://www.cloudera.com/documentation/enterprise/5-12-x/PDF/cloudera-security.pdf

AUTHORS PROFILE

Gousiya Begum is pursuing her PhD (CSE) from JNTUA, Anantapuramu, She has received M.Tech Degree in Computer Science and Engineering from JNTU, Hyderabad. She is currently working as Assistant Professor in Computer Science and Engineering Department in Mahatma Gandhi Institute of Technology, Hyderabad. She is a Life Member of **ISTE**. She has published 7 papers in International Journals and presented 1 paper in International Conference. Her areas of research are Big Data, Data Mining, Natural Language Processing, Network Security, Software Engineering, and Cloud Computing.

Dr. S. Zahoor Ul Huq obtained his Master Degree in Engineering from Anna University and his Ph.D. from Sri Krishnadevaraya University, Anantapur, Andhra Pradesh, India. He is presently working as Professor in the Department of Computer Science and Engineering at G. Pulla Reddy Engineering College(Autonomous), Kurnool, Andhra Pradesh, India. He has presented 32 research papers in various International and National Journals and conferences. His research areas include Cloud Computing, Big Data, Data Analytics and Cloud Computing

Dr. A. P. Siva Kumar, Did his B.Tech from JNTUH, M.Tech from JNTUA, Ph.D from JNTUA in area of "Information Retrieval and Cross Lingual Intelligent Systems" in Year 2011. Recipient of Carrer Award for Young Teachers (CAYT) for the Financial Year 2013-14 with grant of Rs 1,50,000/ by AICTE, New Delhi. Campus Placed from JNTUA in year 2003, and worked in PCS Mumbai and EMC2 Bangalore for three years, Out of intense Passion for Teaching in year 2006 joined JNTUA, Ananthapuramu as Asst.Prof in Department of CSE. He was in Various Admistrative Positions like Deputy Warden, Placement Officer, Addl. Controller of Examinations of JNTUA University, Currently He is Officer In-Charge of Hostels of JNTUA College of Engg. Anantapur. His subjects of interest include Data Analytics, Data Ware Housing and Data Mining, Natural Language Processing, Software Project and Process Management, Software Testing Methodologies, Information Retrieval ,Computer Organization, Operating systems, etc.,

Developed Examination Management Software "JEMS" JNTUA Examination Management System (EMS) which automates various tasks and procedures associated with the pre-examination and the post-examination phases associated with the Examination branch of an Autonomous College. Currently the Software is in live at JNTUACE Ananthapuramu ,JNTUCE Pulivendula, Audisankara College, Gudur and KSRM Kadapa. Also executing one AICTE and one UGC project. Master Trainer of Associate Analytics Trained by Nasscom in association with APSSDC. Very much interested to talk to students, as the same he has been invited by various colleges to talk on topics "Interview Skills" "How to deliver speech effectively" "Inspiring and Motivating " "Presentation Skills", "Change-A perspective in Career" etc.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)