Network Security Enhancement in Hadoop Clusters

Rajesh Laxman Gaikwad¹, Prof. Dhananjay M Dakhane² and Prof. Ravindra L Pardhi³

¹ M. E (Computer Engg.)
Sipna College of Engineering & Technology, Amravati,
Maharashtra, India.

²,³ Department of Computer Science and Engineering
Sipna College of Engineering & Technology, Amravati,
Maharashtra, India.

ABSTRACT

The Apache Hadoop Distributed File System allows companies to store and manage petabytes of data which is collected from disparate data sources and is far more efficient than relational database management systems. A huge number of companies have begun using the open-source technology to aggregate and analyze large volumes of structured and unstructured data which is captured from websites, social media networks, emails, audio and video files, sensors and machines. This paper examines Hadoop cluster and security for Hadoop clusters using Kerberos. Further we see security enhancement using role-based access control, reviewing built-in protections and weaknesses of these systems. The goal is to explore security problems Hadoop data users face with pragmatic advice on how to secure these environments. Since this new technology of data storage is increasing tremendously in many folds, there have been increasing data security and privacy concerns for people who outsource data on Hadoop clusters. There are several ways a user access the data on Hadoop Clusters. This paper explores those methods and also suggests some available advanced methods for scalable, flexible and fine-grained hierarchical access methods for Big data that may be consider for accessing data on Hadoop Clusters.

Keywords: Hadoop Clusters, Data security, access control and Role based access control.

1. INTRODUCTION

The prominent security concerns for Hadoop Clusters is data security and access control because of its Internet based data storage. Hadoop clusters are used as Big data in cloud computing. Here users have to give their data to the cloud servers and most of them is based on Hadoop clusters for storage and business purpose operations. The service providers for Big Data are usually commercial enterprises which cannot be totally entrusted.

Data in Hadoop clusters are important asset for the organization and any user will face serious consequences if its confidentiality is disclosed to their competitive rivals or public. Hadoop supports Cloud computing which is considered as the fifth utility [1] after the four utilities (water, gas, electricity and telephone). The importance of Hadoop storage include reduced costs, capital expenditures, scalability, flexibility and many more benefits. Many service-oriented cloud computing models have been proposed supported by big data that includes Infrastructure as a Service(IaaS), Platform as a Service(PaaS), and Software as a Service(SaaS). Several commercial web service models have been built at different levels, e.g., Amazon’s EC2 [2], Amazon’s S3 [3], and IBM’s blue Cloud [4] are IaaS systems, while Google App Engine [5] and Yahoo Pig are representative PaaS systems, and Google Apps [6] and Salesforce’s Customer Relationship Management (CRM) System [7] belong to SaaS systems. The real benefits of Hadoop storage is that on one hand enterprise users no longer need to invest in hardware/software systems or hire IT professionals to maintain these Computing Systems and hence they save cost on IT infrastructure and human resource where as on other hand, computing system provided by the cloud computing are being offered at a relatively low price in a pay-as-you-use service.

Although Hadoop clusters and Hadoop framework for cloud computing has great benefits for IT industries, but for cloud users and academic researchers think that, security problems in cloud computing will become intolerable, if not handled properly. It will prevent cloud computing real benefits and usage in near future.

This paper is divided into eleven sections and we introduce to reader in section two, present scenario of security in Hadoop clusters, section three deals with Hadoop challenges, section four gives the background of Hadoop, section five describes about Hadoop architecture, section six enlightens us with Security constraints in the world of Hadoop Clusters, section seven discuss security with Kerberos in Hadoop, section eight introduces automation in authentication using Delegation tokens, section nine and ten deals with advanced security models in the form of Security Network Security Enhancement in Hadoop Clusters

Rajesh Laxman Gaikwad¹, Prof. Dhananjay M Dakhane² and Prof. Ravindra L Pardhi³

¹ M. E (Computer Engg.)
Sipna College of Engineering & Technology, Amravati,
Maharashtra, India.

²,³ Department of Computer Science and Engineering
Sipna College of Engineering & Technology, Amravati,
Maharashtra, India.

ABSTRACT

The Apache Hadoop Distributed File System allows companies to store and manage petabytes of data which is collected from disparate data sources and is far more efficient than relational database management systems. A huge number of companies have begun using the open-source technology to aggregate and analyze large volumes of structured and unstructured data which is captured from websites, social media networks, emails, audio and video files, sensors and machines. This paper examines Hadoop cluster and security for Hadoop clusters using Kerberos. Further we see security enhancement using role-based access control, reviewing built-in protections and weaknesses of these systems. The goal is to explore security problems Hadoop data users face with pragmatic advice on how to secure these environments. Since this new technology of data storage is increasing tremendously in many folds, there have been increasing data security and privacy concerns for people who outsource data on Hadoop clusters. There are several ways a user access the data on Hadoop Clusters. This paper explores those methods and also suggests some available advanced methods for scalable, flexible and fine-grained hierarchical access methods for Big data that may be consider for accessing data on Hadoop Clusters.

Keywords: Hadoop Clusters, Data security, access control and Role based access control.

1. INTRODUCTION

The prominent security concerns for Hadoop Clusters is data security and access control because of its Internet based data storage. Hadoop clusters are used as Big data in cloud computing. Here users have to give their data to the cloud servers and most of them is based on Hadoop clusters for storage and business purpose operations. The service providers for Big Data are usually commercial enterprises which cannot be totally entrusted.

Data in Hadoop clusters are important asset for the organization and any user will face serious consequences if its confidentiality is disclosed to their competitive rivals or public. Hadoop supports Cloud computing which is considered as the fifth utility [1] after the four utilities (water, gas, electricity and telephone). The importance of Hadoop storage include reduced costs, capital expenditures, scalability, flexibility and many more benefits. Many service-oriented cloud computing models have been proposed supported by big data that includes Infrastructure as a Service(IaaS), Platform as a Service(PaaS), and Software as a Service(SaaS). Several commercial web service models have been built at different levels, e.g., Amazon’s EC2 [2], Amazon’s S3 [3], and IBM’s blue Cloud [4] are IaaS systems, while Google App Engine [5] and Yahoo Pig are representative PaaS systems, and Google Apps [6] and Salesforce’s Customer Relationship Management (CRM) System [7] belong to SaaS systems. The real benefits of Hadoop storage is that on one hand enterprise users no longer need to invest in hardware/software systems or hire IT professionals to maintain these Computing Systems and hence they save cost on IT infrastructure and human resource where as on other hand, computing system provided by the cloud computing are being offered at a relatively low price in a pay-as-you-use service.

Although Hadoop clusters and Hadoop framework for cloud computing has great benefits for IT industries, but for cloud users and academic researchers think that, security problems in cloud computing will become intolerable, if not handled properly. It will prevent cloud computing real benefits and usage in near future.

This paper is divided into eleven sections and we introduce to reader in section two, present scenario of security in Hadoop clusters, section three deals with Hadoop challenges, section four gives the background of Hadoop, section five describes about Hadoop architecture, section six enlightens us with Security constraints in the world of Hadoop Clusters, section seven discuss security with Kerberos in Hadoop, section eight introduces automation in authentication using Delegation tokens, section nine and ten deals with advanced security models in the form of Security
Enhancement and security using Role-Based Access Control, and last section is about developments in Web authentications for Internet-based users of Hadoop Clusters.

2. PRESENT SCENARIO

It was concluded at Hadoop World Conference that IT executives should pay attention to potential security issues before using Hadoop.[8]

A growing number of firms are using Hadoop and related technologies such as Hive, Pig, and Hbase to analyze data in ways that cannot easily or affordably be done using traditional relational database technologies. JPMorgan Chase use Hadoop to improve fraud detection, IT risk management and self service applications.

Analyt feel that IT operations using Hadoop technology for such applications must be aware of potential security problems. Ebay is using Hadoop technology and Hbase open source database to build a new search engine for its auction site.

It is very difficult to authenticate users and concurrently go about setting up fine-grained access provisions, as said by David Menninger, an analyst with Ventana Research. You can segregate sensitive data into separate files, nodes or clusters, but that still doesn't give you the row-level access control that people are used to in relational databases.

Many government agencies are putting Hadoop-stored data into separate enclaves to ensure that it can be accessed by those with clearance to view it. Several agencies won't put sensitive data into Hadoop databases because of data access concerns.

For users of Hadoop, the most effective security approach is to encrypt data at the individual record level, while it is in transit or being stored in a Hadoop environment

3. HADOOP CHALLENGES

Constant growth - Hadoop clusters are a constantly growing, complex ecosystem of software and provide no guidance to the best platform for it to run on.

Platform decision - The Hadoop community most of the time leave the platform decisions to end users and most of them do not have background in hardware and Lab environments to benchmark all possible design solutions.

Complexity - Hadoop is complex set of software with more than 200 tunable parameters. This parameters are interdependent in tuning and one effects the other in Hadoop environment. They change over the time as job structure changes.

4. HADOOP

Hadoop is a framework for running applications on large clusters built of commodity hardware. The Hadoop framework provides transparently both applications reliability and data mobility.[9]

Hadoop implements a computational paradigm as:

Map/Reduce: Here by Map/Reduce the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster.

HDFS (Distributed File System): HDFS stores data on the compute nodes, providing very high aggregate bandwidth across the cluster. Both Map/Reduce and the distributed file system are designed in such a way that node failures are automatically handled by the framework and thus provide us fault tolerance.

5. HADOOP ARCHITECTURE.

![Figure I: Hadoop Architecture (source: http://hadoop.apache.org)](http://hadoop.apache.org)
The Hadoop Map/Reduce goals are to process large data sets, cope with hardware failure and high throughput. The Map/Reduce daemons are Job Tracker and Task Tracker.

The Hadoop Distributed File System goals are to emphasize streaming data access, store large data sets and cope with hardware failure. HDFS follows master/slave architecture and it has namely Namenode which is master and DataNodes are slaves.

**NameNode (Master):**

An HDFS consists of a single Namenode, a master server that manages the filesystem namespace and regulates access to files by clients. The Namenode makes filesystem namespace operations like opening, closing, renaming etc. of files and directories available via an RPC interface. It also determines the mapping of blocks to Datanodes.

**DataNode (slave):**

This manages storage attached to the nodes that they run on. The Datanodes are responsible for serving read and write requests from filesystem clients, they also perform block creation, deletion, and replication upon instruction from the Namenode.

### 6. SECURITY CONSTRAINTS

Hadoop environments can include data of variety in classifications and security sensitivity concerns. Collection of data into one environment also increases the risk of data theft and accidental disclosure. The technology to collect and store data from multiple sources can create a whole slew of problems related to access control and management as well as data entitlement and ownership[8].

It is felt that, current Access control lists and Kerberos used alone are not adequate for enterprise needs.

Thus it has been observed that because of data Security, access control and lack of role-based access are part of the reason why Hadoop is not ready to replace relational database in the enterprise.

Among these constraints, to address first, we identified the following security risks.[10]

1. Authentication of users is not done by Hadoop services. This is the reason why Hadoop is subject to the following security risks.
   a. Any user can access an HDFS or MapReduce clusters as any other user.
   b. Attackers can masquerade as Hadoop services

2. There is no enforcement of access control by DataNodes on accesses to its data blocks.

   The fact behind above security lapses is Hadoop only provides the file permissions system in HDFS with the purpose of preventing one user from accidentally wiping out the whole filesystem from a bug in a program, or by mistakenly typing hadoop fs -rmr /, but it doesn’t prevent a malicious user from assuming root’s identity to access or delete any data in the cluster [11].

   From the chapter of security, what was missing was a secure authentication mechanism to assure Hadoop that the user seeking to perform an operation on the cluster is who they claim to be and therefore trusted. HDFS file permissions provide only a mechanism for authorization, which controls what a particular user can do to a particular file. For example, a file may only be readable by a group of users, so anyone not in that group is not authorized to read it. However, authorization is not enough by itself, since the system is still open to abuse via spoofing by a malicious user who can gain network access to the cluster.

   This was what the issue Yahoo! faced in 2009, which led a team of engineers there to implement secure authentication for Hadoop. In their design implementation, Hadoop itself does not manage user credentials, since it relies on Kerberos, a mature open-source network authentication protocol, to authenticate the user. In turn, Kerberos doesn’t manage permissions. Kerberos says that a user is who they say they are; it’s Hadoop’s job to determine whether that user has permission to perform a given action.

### 7. KERBEROS

Kerberos is a network authentication protocol developed at MIT as part of Project Athena. It uses private-key cryptography for providing authentication across open networks. It was developed before the popularity of public key cryptography and systems like SSL. While many systems today use public key cryptography for authentication, Kerberos manages to do it with symmetric key cryptography.
In all, there are three steps that a client must take to access a service when using Kerberos, each of which involves a message exchange with a server:

1. **Authorization.** The client authenticates itself to the Authentication Server and receives a timestamped Ticket-Granting Ticket (TGT).
2. **Authorization.** The client with the TGT request a service ticket from the Ticket Granting Server.

![Figure II: The three-step Kerberos ticket exchange protocol](image)

3. **Service Request.** The client uses the service ticket to authenticate itself to the server that is providing the service the client is using. In the case of Hadoop, this might be the namenode or the jobtracker. Together, the Authentication Server and the Ticket Granting Server form the **Key Distribution Center (KDC)**. The process is shown graphically in figure II.

The authorization and service request steps are not user-level actions. Here the client performs these steps on the user’s behalf. The authentication step, however, is normally carried out explicitly by the user using the kinit command, which will prompt for a password. However, this doesn’t mean you need to enter your password every time you run a job or access HDFS, since TGTs last for 10 hours by default (and can be renewed for up to a week). It’s common to automate authentication at operating system login time, thereby providing **single sign-on** to Hadoop.

In cases where you don’t want to be prompted for a password (for running an unattended MapReduce job, for example), you can create a Kerberos **keytab** file using the ktutil command. A keytab is a file that stores passwords and may be supplied to kinit with the -t option.

raj@laptop:~$ kinit
Password for raj@BOB.COM:
raj@laptop:~$ ssh work
Welcome to work!
...do stuff...
raj@work:~$ exit
raj@laptop:~$ mailx
...read e-mail...
raj@laptop:~$ klist
Ticket cache: FILE:/tmp/krb5cc_201
Default principal: raj@BOB.COM
Valid starting Expires
12/13/05 23:07:25 12/14/05 07:07:25
Service principal krbtgt/BOB.COM@BOB.COM
renew until 12/14/05 23:07:24
12/13/05 23:11:57 12/14/05 07:07:25
Service principal host/work.bob.com@BOB.COM
renew until 12/14/05 23:07:24
12/13/05 23:33:05 12/14/05 07:33:03
Service principle imap/mail.bob.com@BOB.COM.
Limitation of Kerberos: Kerberos can only provide authentication while other protocols (such as NIS or LDAP) are still needed for authorization. All applications must be “Kerberized” to take advantage of cryptography. Kerberos provides standard APIs to authenticate your applications. There are some PAM modules for Kerberos authentication. We cannot migrate existing password hashes into the Kerberos database. So the authentication is only as good as the user's password and it assumes relatively secure hosts on an insecure network.

8. DELEGATION TOKENS

In a system like HDFS or MapReduce, there are many client-server interactions, each of which must be authenticated. For instant, an HDFS read operation will involve multiple calls to the namenode and calls to one or more datanodes. To authenticate each call, instead of using the three-step Kerberos ticket exchange protocol, which would present a high load on the KDC on a busy cluster, Hadoop uses delegation tokens to allow later authenticated access without having to contact the KDC again. On behalf of users delegation tokens are created and used transparently by Hadoop, so there’s no action you need to take as a user over using kinit to sign in.

9. SECURITY ENHANCEMENTS REQUIRED

Using the operating system account tasks can be run for the user who submitted the job, rather than the user running the tasktracker. This means that the operating system is used to isolate running tasks, so they can’t send signals to each other (to kill another user’s tasks), and so local information, such as task data, is kept private via local file system permissions. This feature is enabled by setting mapred.tasktracker.task-controller to org.apache.hadoop.mapred.LinuxTaskController.* In addition, administrators need to ensure that each user is given an account on every node in the cluster (typically using LDAP).

• It is found tasks that are run as the user who submitted the job, the distributed cache is secure: files that are world-readable are put in a shared cache (the insecure default), otherwise they go in a private cache, only readable by the owner.
• Users can view and modify only their respective jobs, not others. This is enabled by setting mapred.acls.enabled to true. There are two job configuration properties, mapreduce.job.acl-view-job and mapreduce.job.acl-modify-job, which may be set to a comma-separated list of users to control who may view or modify a particular job.
• Also the shuffle is secure, preventing a malicious user from requesting another user’s map outputs.

10. SECURITY USING ROLE-BASED ACCESS CONTROL

Bell-La Padula (BLP) [12] and BiBa [13] are two world class security models. To achieve flexible and fine-grained access control of schemes [14]-[17] have been proposed more recently. But these schemes are only applicable to systems in which data owners and service providers are within the same trusted domain. As far cloud computing is concerned data owners and service providers are usually not in same trusted domain.

A new access control scheme employing attribute-based encryption [18] was proposed by V.Goyal and his team. Yu et al. [19], which adopts the so-called key - policy attribute based encryption (KP-ABE) to enforce fine-grained access control however, fails short of flexibility in attribute management and lacks scalability in dealing with multiple levels of attribute authorities. It has been observed that in contrast to KP-ABE, ciphertext-policy ABE (CP-ABE) [20] turns out to be well suited for access control policies.

A user is able to decrypt a cipher-text only if there is a match between his decryption key and cipher-text. CP-ABE is conceptually closer to tradinational access control models such as Role-Based Access Control (RBAC) [20]. Thus it is more natural to apply CP-ABE, instead of KP-ABE, to enforce access control of encrypted data. However basic CP-ABE schemes cannot provide enough support to access control in modern enterprise environments, which should have considerable flexibility and efficiency in specifying policies and managing user attributes. In a CP-ABE scheme, decryption keys only support user attributes that are organized logically as a single set, so users can only use all possible combinations of attributes in a single set issued in their keys to satisfy policies. To solve this problem, Bobba et al [21]., introduced ciphertext-policy attribute-set-based encryption (CP-ASBE or ASBE for short). ASBE is an extended form of CP-ABE which organizes user attribute into a recursive set structure. ASBE can enforce dynamic constraints on combining attributes to satisfy a policy, which provides great flexibility in access control. The missing part of ASBE is the delegation algorithm, which is used to construct the hierarchical structure. We adopt four algorithms of ASBE(Setup, KeyGen, Encrypt, Decrypt) and extend ASBE by proposing a new delegation algorithm.

As proposed by Wang et al. [22] Hierarchical attribute-based encryption (HABE) to achieve fine-grained access control in cloud storage services by combining hierarchical identity based encryption (HIBE) and CP-ABE. This scheme also supports fine-grained access control and fully delegating computing to the cloud providers. However, HABE uses disjunctive normal form policies and assumes all attributes in one conjunctive clause are administrated by the same domain master. Thus the same attribute may be administrated by multiple domain masters according to specific policy,
which is difficult to implement in practice. Furthermore compared to ASBE, this scheme cannot support compound attributes efficiently and does not support multiple value assignments.

Another proposed scheme HASBE [23] seamlessly extends the ASBE scheme to handle the hierarchical structure of systems users in figure 1.

![Hierarchical structure of user in the system.](source: HASBE: Z. Wan, Jun’e Liu, and Robert H. Deng, Senior Member, IEEE)

This system model consists of trusted multiple domain authority, and numerous user corresponding to data owners and data consumers. The trusted authority is responsible for generating and distributing system parameters and root master keys as well as authorizing the top level domain authorities. A domain authority is responsible for delegating keys to subordinate domain authority at the next level or user in its domain. Each user in the system domain is assigned a key structure which specifies the attributes associated with the user’s decryption Key.

11. WEB USER INTERFACE AUTHENTICATION

According to CDH3u5[24], Security for web user interface in Hadoop is the Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO), sometimes referred to as "spen-go". Prior to CDH3u5, Kerberized SSL (KSSL) was the only option for performing HTTP authentication. In versions of CDH3u5 and later, you may choose to use either SPNEGO or KSSL to perform HTTP authentication. In order to maintain backward compatibility with secure clusters prior to CDH3u5, the default configuration is still to use KSSL. However, Cloudera strongly recommends that you follow the instructions manual to use SPNEGO for the following reasons:

- Because of a bug in JDK 6, KSSL does not work with Kerberos tickets that use cryptographically strong encryption types. This requires you to use weak encryption types, thus lowering the theoretical security of the system.

- SPNEGO is the recommended HTTP authentication mechanism for WebHDFS introduced in CDH3u5. If you use SPNEGO for WebHDFS authentication, then you will already have created the necessary Kerberos principals to support HDFS checkpointing using SPNEGO.

- Other projects like Oozie and Sqoop in the Hadoop ecosystem use SPNEGO for HTTP authentication.

- It is found that, the error messages that one receives are not very helpful and debugging KSSL misconfigurations is very difficult.

12. CONCLUSION

This paper has explored the pros and cons of using Big Data in the form of Hadoop Clusters. This is technology challenge for the future of Cloud Computing. There are many areas to improve in the various security aspects of Hadoop clusters and new technologies are proposed to ensure the security in terms of reliability and flexibility. Still lot of work is remaining to make Hadoop Clusters as Full fledged Database system in terms of user accountability and dynamic data updating. These two fields open new avenues for research and interesting work can be proposed for third party auditor for existing and new security models in Hadoop clusters.

13. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the editor and all those who help to review and give their valuable suggestions that significantly improved the quality of this paper.
REFERENCES


Authors

Rajesh L Gaikwad. Currently working as a lecturer in Viva College and pursuing M.E. (Computer Engg) from Sipna COE&T, Amravati which is affiliated to SGB Amravati University. The Area of Interest is Network Security.

Prof. Dhananjay M. Dakhane has done M.E in Computer Engg. and pursuing Ph. D. He is an Associate Professor with 14 years of experience in Sipna COE&T, Amravati which is affiliated to SGB Amravati University. The Area of Interest is Network Administration and Security, Operating Systems.

Prof. Ravindra L. Pardhi has done M.E. (Information Technology ). He is an Assistant Professor with 5 Years of experience in Sipna COE&T, Amravati which is affiliated to SGB Amravati University. His area of interest is Web Technology, Object Oriented Programming.