COMPREHENSIVE AND COORDINATED SECURITY OF KNOX GATEWAY IN BIG DATA

M.Chithik Raja[1] , M. Munir Ahamed Rabbani[2]

[1]Research Scholar , AMET University , (India)
[2]Supervisor, B. S. Abdur Rahman University , (India)

ABSTRACT

Adding security to Hadoop is challenging because all the interactions do not follow the classic client-server pattern: the file system is partitioned and distributed requiring authorization checks at multiple points; a submitted batch job is executed at a later time on nodes different from the node on which the client authenticated and submitted the job; job tasks from different users are executed on the same compute node; secondary services such as a workflow system access Hadoop on behalf of users; and the system scales to thousands of servers and tens of thousands of concurrent tasks. To address these challenges, the base Kerberos authentication mechanism is supplemented by delegation and capability-like access tokens and the notion of trust for secondary services.In order to secure a Hadoop cluster all you needed was a firewall that restricted network access to only authorized users. This eventually evolved into a more robust security layer in Hadoop layer that could augment firewall access with strong authentication. The enterprise has placed requirements on the platform to not only provide perimeter security, but to also integrate with all types of authentication mechanisms and all the while, be easy to manage and to integrate with the rest of the secured corporate infrastructure. Kerberos can still be a great provider of the core security technology but with all the touch-points that a user will have with Hadoop, something more is needed.

Keywords: Git Hub EDW, GFS, HBASE, GCE, HDFS

I INTRODUCTION

Hadoop comes from open source, and there's no shortage of open source Apache projects aimed at building security functionality into the core Hadoop stack. A quick list would include Apache Knox for authentication, Apache Falcon for data governance, and Apache Sentry, for role-based authorization. We take Kerberos for authentication, while hart on network gave us open source Project Rhino, a GitHub-hosted effort to develop a Hadoop framework for encryption, key management, and authorization.But stitching these components together in a comprehensive and cohesive way is not so easy. Apache Sentry, for example, requires working with XML files, which is not something everybody will feel comfortable with. Many of these open source projects are quite green still and lack polish.This

open source Hadoop security has provided room for third-party vendors to innovate, and has also driven the major distributors to take matters into their own hands.

II.HADOOP BACKGROUND

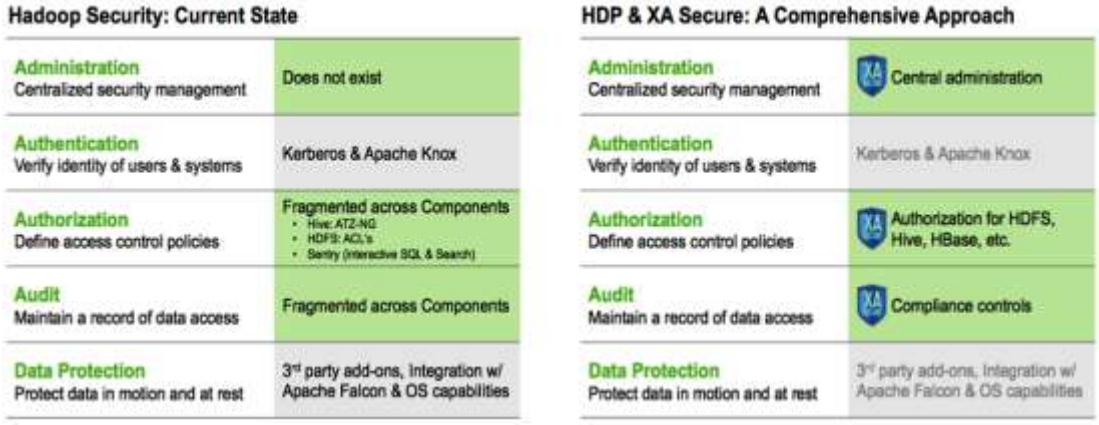


Figure.1 Hadoop Security approach

Hadoop has been under development at Yahoo! and a few other organization as an Apache open source project over the last 5 years. It is gaining wide use in the industry. Yahoo!, for example, has deployed tens of Hadoop clusters, each typically with 4,000 nodes and 15 petabytes. Hadoop contains two main components. The first component, HDFS, is a distributed file system similar to GFS. HDFS contains a metadata server called the NameNode that stores the hierarchical file and directory name space and the corresponding metadata, and a set of DataNodes that stores the individual blocks of each files. Each block, identified by a block id, is replicated at multiple DataNodes. Client perform file metadata operations such as create file and open file, at the NameNode over an RPC protocol and read/write the data of a file directly to DataNodes using a streaming socket protocol called the data-transfer protocol

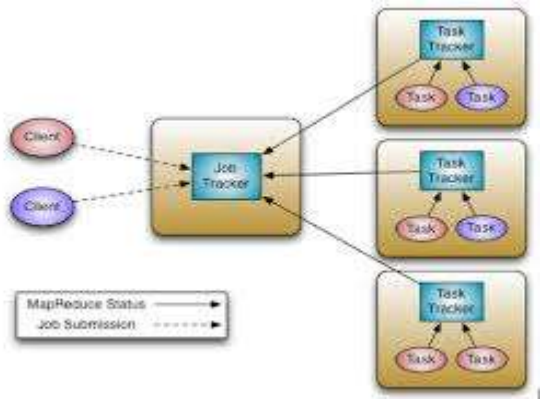


Figure 2.Hadoop higher level architecture

The second component is a framework for processing large amounts of data in parallel using the MapReduce paradigm . HDFS DataNodes also serve as compute nodes for MapReduce to allow computation to be performed close to the data being processed. A user submits a MapReduce job to the JobTracker which schedules the job to be

executed on the compute nodes. Each compute node has a small daemon called the TaskTracker that launches map and reduce tasks of a job; the task tracker also serves intermediate data created by map tasks to reduce tasks. There are additional services deployed with Hadoop, one of which are relevant to the security concerns discussed in this paper. Ooozie is a workflow system that provides a way to schedule and submit a DAG of MapReduce jobs that are triggered for execution by data availability or time.

III. FLEXIBILITY AND PERFORMANCE

Ability to access and store various types of data – both structured and unstructured, with no constraints of schema-on-write, along with the emergence of new ways of accessing and processing data – e.g. Storm for real-time/streaming data, SQL-like tools including Impala,, Stinger etc.Due to these key advantages, Hadoop lends itself to several data processing use cases.

- Data store / Enterprise data warehouse (EDW) – cost-effective storage for all of an organization's ever expanding data
- Active archive – allowing cost-effective querying on historical data from archival systems
- Transformation – executing data transformations for improved throughput and performance
- Exploration – allows fast exploration and quicker insights from new questions and use cases, taking advantage of Hadoop's schema-on-read model instead of schema-on-write models of traditional relational databases
- Real-time applications – usage of flexible add-ons like Storm to provide dynamic data mashups
- Machine learning, data mining, predictive modeling and advanced statistics
- The early adopters of Hadoop are the web giants like Facebook, Yahoo, Google, LinkedIn, Twitter etc.

Facebook uses Hadoop – Hive and HBase for data warehousing (over 300 PB in aggregate and over 600 TB daily data inflows) and real-time application, serving up dynamic pages customized for each of its over 1.2 billion users.

Yahoo uses Hadoop and Pig for data processing and analytics, web search, email antispam and ad serving with more than 100,000 CPUs in over 40,000 servers running Hadoop with 170 PB of storage .

Google had used MapReduce to create its web index from crawl data and also uses Hadoop clusters on its cloud platform with Google Compute Engine (GCE).

LinkedIn uses Hadoop for data storage and analytics driving personalized recommendations like "People you may know" and ad targeting.

Twitter uses Hadoop – Pig and HBase for data visualization, social graph analysis and machine learning.

IV.LIMITATIONS OF HADOOP

While Hadoop is the most well-known Big Data solution, it is just one of the components in the Big Data landscape. While in theory, Hadoop is infinitely scalable and resilient and allows a great deal of flexibility in storing structured and unstructured data, in practice, there are several considerations to be taken care of while architecting Hadoop clusters due to the inherent limitations of Hadoop.Hadoop should not be a stand-alone solution, else it will quickly become a data silo unconnected with the rest of the data management infrastructure. The Hadoop strategy needs to fit into the overall data management and processing framework of the organization to allow for growth and maintenance while not sacrificing on flexibility and agility.

In the enterprise, security is a big deal. While Hadoop was originally built without a security model, the Hadoop ecosystem is evolving with various projects for security, including Kerberos authentication, the Sentryoffering from Cloudera, Project Rhino etc.

V.SECURITY IN THE KNOX GATEWAY

The Knox Gateway ("Gateway" or "Knox") is a system that provides a single point of authentication and access for Apache Hadoop services in a cluster. The goal is to simplify Hadoop security for both users (i.e. who access the cluster data and execute jobs) and operators (i.e. who control access and manage the cluster). The Gateway runs as a server (or cluster of servers) that serve one or more Hadoop clusters. It has few key functions:

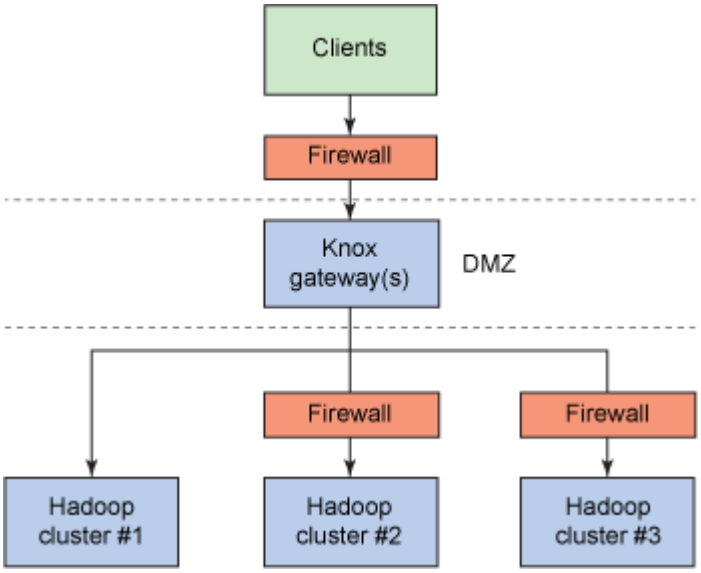
- Provide perimeter security to make Hadoop security setup easier
- Support authentication and token verification security scenarios
- Deliver users a single cluster end-point that aggregates capabilities for data and jobs
- Enable integration with enterprise and cloud identity management environments
- Manage security across multiple clusters and multiple versions of Hadoop

Knox will be able to provide a security layer for multiple clusters and multiple versions of Hadoop simultaneously and will deliver a simple intuitive management interface. Playing nice with others is always a security imperative, so Knox will integrate with the existing frameworks for Active Directory /LDAP and it will allow for extensions for custom authentication mechanisms

Intel, Apache Knox as reverse proxy or using Apache Accumulator for cell-level security; however most are complex to setup and there is still no reference standard across deployments as problems with Map and Reduce steps and need other data processing paradigms. There are improvements being developed with Storm for real-time access or Spark for improving the data analytics performance with in-memory distributed computing to get around these

issues. In the next parts of this series, I will explore topics of building a Hadoop data warehouse, big data analytics with tools like R as well as other Big Data solutions, Hadoop enhancements and alternatives to Hadoop.

VI.PERIMETER SECURITY WITH THE APACHE KNOX GATEWAY



The Apache Knox Gateway provides a perimeter security solution for Hadoop. Where Sentry provides fine-grained access controls to data, the Knox Gateway provides controlled access to Hadoop services. The goal of the Knox Gateway is to provide a single point of secure access for Hadoop clusters. The solution is implemented as a gateway (or small cluster of gateways) that exposes access to Hadoop clusters through a Representational State Transfer (REST)-ful API. The gateway provides a firewall between users and Hadoop clusters (see Figure 2) and can manage access to clusters that run different versions of Hadoop.

The Knox Gateway is a complementary security solution to Sentry that provides the outer level of access security. As a gateway in a demilitarized zone, Knox Gateway provides controlled access to one or more Hadoop clusters segregated by network firewalls.

6.1 Delegation tokens

Where the Apache Knox Gateway and Sentry provide perimeter and data access security, one missing element is HDFS data access from MapReduce tasks. One solution used by Oozie relies on the concept of delegation tokens. A **delegation token** is a two-party authentication protocol that lets users authenticate themselves with the Namenode (using Kerberos); on receipt of the delegation token, users can provide the token to the JobTracker so that resulting Hadoop jobs for token provided for future Hadoop work accesses the HDFS. Any resulting MapReduce tasks for the job uses the associated delegation token to fully secure the resulting work.Delegation

tokens rely on a two-party authentication that is simpler and more efficient than the three-party authentication that Kerberos uses. This difference minimizes Kerberos traffic and leads to improved scaling and minimizing load on the Kerberos assets.A user submitting a job authenticates with the Job Tracker using Kerberos. The job is executed later, possibly after the user has disconnected from the system. How do we propagate the credentials? There are several options:

- have the user pass the password to the job tracker
- pass the Kerberos credentials (TGT or service ticket for the NameNode)
- use a special delegation token Passing the password is clearly unacceptable. We choose to use a special delegation token instead of passing the Kerberoscredentials .

After initial authentication to NameNode using Kerberos credentials, a client obtains a delegation token, which is given to a job for subsequent authentication to NameNode. The token is in fact a secret key shared between the client and NameNode and should be protected when passed over insecure channels. Anyone who gets it can impersonate as the user on NameNode. Note that a client can only obtain new delegation tokens by authenticating using Kerberos. The format of delegation token is:

TokenID = {ownerID, renewerID, issueDate,

maxDate, sequenceNumber}

When a client obtains a delegation token from NameNode, it specifies a renewer that can renew or cancel the token. By default, delegation tokens are valid for 1 day from when they are issued and may be renewed up to a maximum of 7 days. Because MapReduce jobs may last longer than the validity of the delegation token, the JobTracker is specified as the renewer. This allows the JobTracker to renew the tokens associated with a job once a day until the job completes. When the job completes, the Job Tracker requests the NameNode to cancel the job's delegation token. Renewing a delegation token does not change the token, it just updates its expiration time on the NameNode, and the old delegation token continues to work in the MapReduce tasks. The NameNode uses a secret key to generate delegation tokens; the secret is stored persistently on the NameNode. The persistent copy of the secret is used when the NameNode restarts. A new secret is rolled every 24 hours and the last 7 days worth of secrets are kept so that previously generated delegation tokens will be accepted. The generated token is also persistently.

VII. MAPREDUCE

A MapReduce job involves the following stages

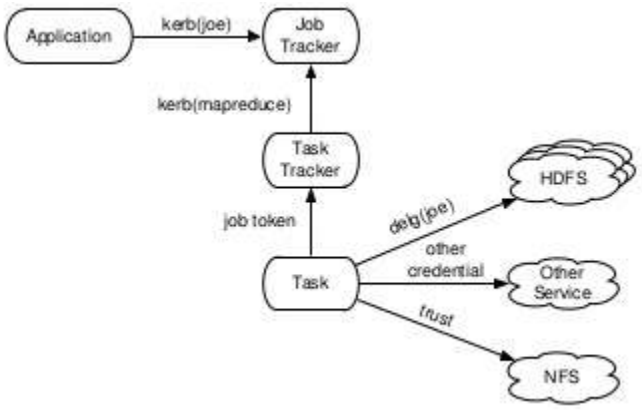


Figure 3. Map Reduce Authentication

1. A client connects to the JobTracker to request a job id and an HDFS path to write the job definition files. The MapReduce library code writes the details of the job into the designated HDFS staging directory and acquires the necessary. HDFS delegation tokens. All of the job files are visible only to the user, but depend on HDFS security.

2. The JobTracker receives the job and creates a random secret, which is called the job token. Thejob token is used to authenticate the job's tasks to the MapReduce framework.

3. Jobs are broken down into tasks, each of represents a portion of the work that needs to be done. Since tasks (or the machine that they run on) may fail, there can be multiple attempts foreach task. When a task attempt is assigned to a specific TaskTracker, the TaskTracker creates a secure environment for it. Since tasks from different users may run on the same compute node, we have chosen to use the host operating system's protection environment and run the task as the user. This enables use of the local file system

and operating system for isolation between users in the cluster. The tokens for the job are stored in the local file system and placed in the task's environment such that the task process and any sub-processes will use the job's tokens.

4. Each running task reports status to its Task- Tracker and reduce tasks fetch map output from various TaskTrackers. All of these accesses are authenticated using the job token.

5. When the job completes (or fails), all of the HDFS delegation tokens associated with the job are revoked.

7.1 Job Submission

A client submitting a job or checking the status of a job authenticates with the JobTracker using Kerberosover RPC. For job submission, the client writes the job configuration, the job classes, the input splits,and the meta information about the input splits into a directory, called the Job Staging directory, in theirhome directory. This directory is protected as read, write, and execute solely by the user.

HDFS and other services. Therefore, the job needs to package the security credentials in a way that a task can later look up. The job's delegation tokens are keyed by the NameNode's URL. The other credentials, for example, a username/password combination for a certain HTTP service are stored similarly. The client then uses RPC to pass the location of the Staging directory and the security credentials to the JobTracker. The JobTracker reads parts of the job configuration and store it in RAM. In order to read the job configuration, the JobTracker uses the user's delegation token for HDFS. The JobTracker also generates a random sequence of bytes to use as the job token, When the job is submitted, the JobTracker creates a secret key that is only used by the tasks of the job when identifying themselves to the framework. As mentioned earlier, this token is stored as part of the credentials file in the JobTracker's system directory on HDFS. This token is used for the RPC via DIGEST-MD5 when the Task communicates with the TaskTracker to requests tasks or report status. Additionally, this token is used by Pipes tasks, which run as sub-processes of the MapReduce tasks. Using this shared secret, the child and parent can ensure that they both have the secret. The security credentials passed by the client, and, the job token are stored in the JobTracker's system directory in HDFS, which is only readable by the'MapReduce' user. To ensure that the delegation tokens do not expire, the JobTracker renews them periodically. When the job is finished, all of the delegation tokens are invalidated.

VIII.CONCLUSION

Hadoop's central authentication mechanism is Kerberos; it is supplemented by delegation tokens,capability-like access tokens and the notion of trust for auxiliary services. The delegation token addressesthe problem of propagating the user's credentials to an executing job. We compared the mechanisms used to generate, exchange, renew and cancel the delegation token with those of Kerberos It was a good decision to not dothat as it would have been challenging to have a modifiedKerberos server adopted for non-Hadoop use inour organization. Further it would have hindered thewider-adoption of secure Hadoop in the industry. Our security mechanisms,especially in the Delegation Token layer, are designed to be pluggable;we believe it should be fairly easy to, say replace,our primary authentication mechanism Kerberoswith another mechanism. Our approach hasthe advantage that one could continue to use our tokensto supplement a different primary authenticationmechanism. However as Hadoop's use grew at Yahoo!, security became critical. Hadoop's elastic allocation of resources and the scale at which it is typically deployed lends Hadoop to be shared across tenants where security is critical if any stored data is sensitive. Segregating sensitive data and customers into private clusters was not practical or cost effective.

IX.REFERENCE

[1].N. Afrati and Jeffrey D. Ullman. Optimizing Multiway Joins in a Map- Reduce Environment. IEEE Transactions on Knowledge and Data Engineering, 23 (9):1282 – 1298, 2011.

[2].EytanBakshy, Jake M Hofman, Duncan J Watts, and Winter A Mason. Everyoneâ ˘A Z´ s an Influencer : Quantifying Influence on Twitter. In WSDM '11: 4th International Conference on Web Search and Data Mining, WSDM '11, pages 65– 74, 2011.

[3]. Chickowski, "A Case Study in Security Big Data Analysis," Dark Reading, 9 Mar. 2012.

[4]. François et al., "BotCloud: Detecting Botnets Using MapReduce," Proc. Workshop Information Forensics and Security, IEEE, 2011, pp. 1–6.

[5]. P. Giura and W. Wang, "Using Large Scale Distributed Computing to Unveil Advanced Persistent Threats," Science J., vol. 1, no. 3, 2012, pp. 93–105.

[6]. N. Pansare, V. R. Borkar, C. Jermaine, and T. Condie. Online aggregation for large mapreduce jobs.PVLDB, 2011.

[7].Konstantin Shvachko, HairongKuang,SanjayRadia, and Robert Chansler. The Hadoop Distributed File System. Mass Storage Systems and Technologies, IEEE / NASA Goddard Conference on, 0:1–10,2010.

[8].Mario Cataldi, Università Torino, Luigi Di Caro, Claudio Schifanella, and Luigi Di Caro. Emerging Topic Detection on Twitter based on Temporal and Social Terms Evaluation. In MDMKDD '10: 10th InternationalWorkshop on Multimedia Data Mining, pages 4:1—-4:10, 2010.

[9].http:/www.nature.com/nature/journal/vaop/ncurrent/full/nature11875.html

[10].T.-F. Yen et al., "Beehive: Large-Scale Log Analysis for Detecting Suspicious Activity in Enterprise Networks," to be published in Proc. Ann. Computer Security Applications Conference (ACSAC 13), ACM, Dec. 2013.

[11].Zach Miller, Dan Bradley, Todd Tannenbaum, and Igor Sfiligoi. Flexible session management in a distributed environment. Journal of Physics: Conference Series, 219(4):042017, 2010.