

Big Data Authentication and Authorization in HDP (Hadoop Distributed platform) using Kerberos and Ranger

Chandni Grover¹, Manpreet Kaur Aulakh²

^{1,2}Department of Computer Science and Engineering,

Shaheed Bhagat Singh State Technical Campus Ferozepur (India)

ABSTRACT

In this paper, we propose a solution for user and services authentication and authorization in Hadoop Distributed Platform (HDP). We have implemented MIT Kerberos for Authentication in an automated fashion that involves creating Principals and give it to Ambari with admin privileges and Ambari will control all the services, clients and keytabs. After enabling Kerberos we can access the HDFS by using our Kerberos key using kinit and access the file system and Apache Ranger for consistent Authorization control across all apache components and file system with in HDP.

Keywords: *Authentication, Authorization, Kerberos, Apache Ranger, Security, Big Data, JCE*

I. INTRODUCTION

Today Big Data privacy and security is the biggest concern that revolves around the protection of Sensitive Data and information. [1] As the Data is increasing day by day so as the risks associated with the Big data and it becomes important to provide effective security measures to data. Apache Hadoop is one of the most popular platforms for Big Data processing using commodity hardware for data analysis and data processing. Organizations built up their own clusters to manage the data privacy and Security [7],[10]. Security protocols [1], [2] are used for defining the rules for communication between the users, servers and applications. For providing Security at software level in Hadoop Distributed platform for providing access to the users and groups/clients to access the Hadoop services, we are using Kerberos as an authentication framework. As far the research Kerberos is the primary authentication protocol in the hadoop that assures that user is connecting to the trusted server while entering its credentials (username and password)

II. SECURITY ISSUES IN HADOOP CLUSTER

2.1 Authentication

It is defined as checking the identity of a participant by checking the username and password. [3,9]

2.2 Authorization

It is used to check whether the participant is allowed to perform an action or if the user authorized to run that particular service. This is done by checking the excess control list or checking the credentials whether the user can access the specific resource/server/Name node.

2.3 Auditing

What exactly the user is doing on your system whether he is adding new data or deleting or modifying the content. Auditing is also implemented by an individual user, and events will be written down into local files.

2.4 Data Integrity

If the data go from one node to another node you want to be sure that the data left as X and should receive as X on the other end. [9]

2.5 Confidentiality

When somebody wants to access the data, the data should be made available only to the authorized user. Data is kept in encrypted format and only available to the authorized user. Hadoop uses TDE Transmitted Data Encryption. [9]

III. BACKGROUNDS

Before Diving into the Details for implementing the Solution for Authentication and Authorization, for Convenience there are some basics to introduce, mainly about Kerberos and Apache Ranger

3.1 Kerberos

Kerberos is an Authentication protocol for trusted hosts on Untrusted Network [5]. The trusted hosts mean these hosts that are kerberized need to belong to a particular Realm. Authentication is done by using a central server. It works on the principal of Single Sign on as the user is asked to enter a password only once per work session. Kerberos is highly time dependent so all the clients Clocks must be Synchronized

Terms used in Kerberos:

3.1.1 REALM

Realm is collection of Principal than belong to the same domain. REALM is case sensitive always written in capital letters.

3.1.2 Principal

Principal is any entry in Kerberos database. It could be a User, Service and Server. E.g.

Name/Instance@REALM. In our case we are using admin/admin@HORTONWORKS.COM

3.1.3 KDC

It stands for Key Distribution Centre. Its components are:

A. Database

It stores the principals.

B. Authentication server (AS)

This is responsible for authenticating the users.

C. Ticket granting server (TGS)

It is responsible for providing Ticket.

3.1.4 Ticket

Ticket allows you to access some particular service. Client presents the ticket to application server to demonstrate the authenticity of its identity.

3.1.5 Application Server

It is a server that is running a particular service that we want to access, e.g. IMAP server, SSH server which is running a service which is Kerberized.

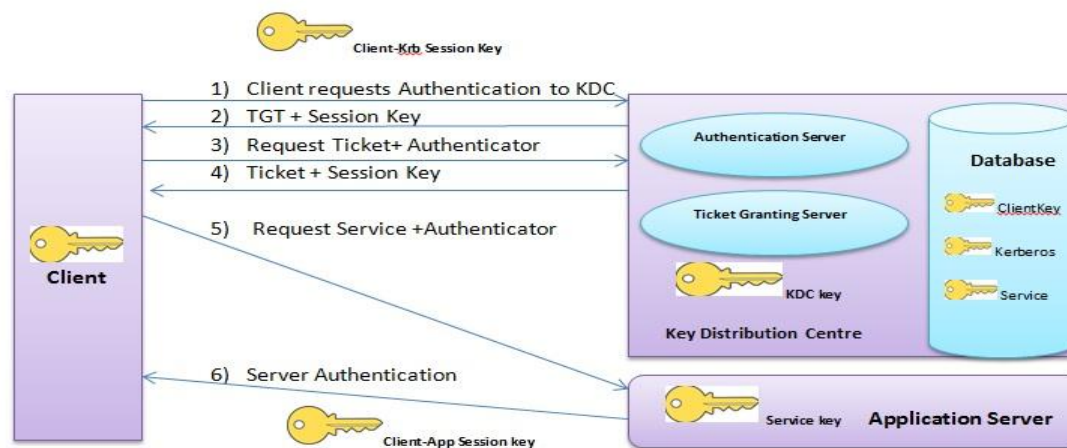


Figure: Kerberos Key Distribution Centre with its Components

Fig.1 Kerberos Key Distribution Centre and its Components

3.2 Steps Involved in Kerberos Authentication Process:

3.2.1 TGT Generation

The Client requests the Authenticator Server for the TGT. The KDC checks the entry for the Client in its Database. If the Credentials matched, the KDC generates a TGT for the client. A Client-Krb session key is created and sent back to the Client with TGT which is encrypted by the Private Key of the KDC and Client Private Key. If the client has the credentials it decrypts the private key.[4]

3.2.2 TGS Session Ticket Generation

The Client uses TGT and sends a request for the Ticket. The TGT provides a Ticket+ Client-Krb Session Key to the Client for Authentication.

3.2.3 Service Access

Now, if the Client wants to access the service of the Application server e.g. ssh service. It sends an Authenticator (User Name, Ip and Time Stamp) + Session Key + TGT to the TGS. The TGS fetches the Authenticator + TGT and has the Client-krb session key, it decrypts the packet. The Authenticator Credentials (User Name, Ip and Time Stamp) are matched by the KDC. If the credentials get matched the Ticket is sent to Client which is encrypted by the private key of the Application Server and client-krb session key and Client – app Session key. The Client can decrypt this Ticket and access the service.

3.3 Ranger Architecture

3.3.1 COMPONENTS

A. Ranger policy Admin Server

It is a web base console allows user to put their policy there and policies get stored in Ranger policy database.[6]

B.Features

- [1] Provides web interface to support Ranger activities. Like Define repository, Define excess policy, Define auditing policies, Manage user/group, View/ Analyze and Audit data.
- [2] Run embedded Tomcat server
- [3] Supports LDAP/AD and authentication to get into this tool.

C. Ranger User/Group Sync Server

When we define any security policy we define specific set of users or groups. And while writing these users or groups we do not want to do any spelling mistakes or type it. We have to be sure that if the permission is given to user X, it has to be X. So to do that we go and get all the users and group that are in our corporate directory either LDAP or AD or UNIX server authentication as a resource for user or group. We take all the user and group information and synchronize them back to Ranger Database, so that write user and groups are selected.

D. Features

- [1] Standalone Java Server.
 - a) Retrieve user/groups from enterprise directories.
 - b) Supporting policies definition.
 - c) Allowing excess to Ranger policy Admin server.
- [2] Support synchronization of user/groups
 - a) LDAP support
 - b) Active Directory

E. Ranger Plugins

The plugins are Light Weight Java components that exist between your hadoop component to enforce the ACL and generate Audit logs and put them into the Centralize audit log. Ranger plugins act as an authorizer with in Name node. After we define ranger as the authorizer for HDFS user can go to Ranger policy admin server and define

- [1] Policies(read,write,execute)on files and folders.
- [2] Use of wild cards to define policies.

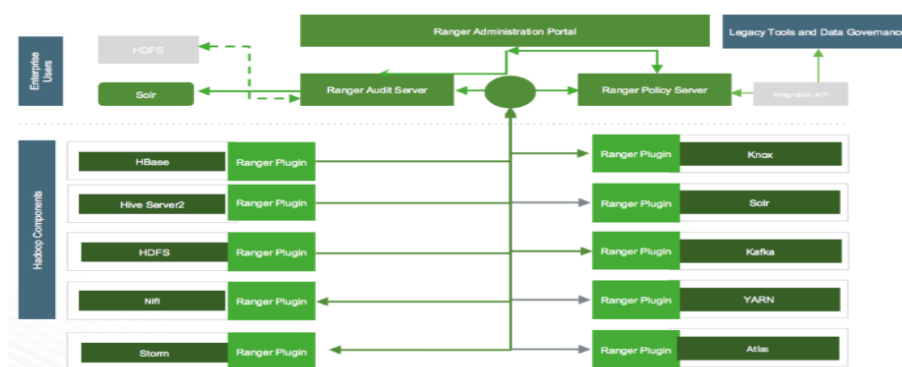


Fig.2 Apache Ranger Architecture

VI. INTRODUCING AUTHENTICATION AND AUTHORIZATION FOR HDFS

4.1 Kerberos and its Implementation

When we are trying to authenticate against a particular Server or against specific credentials we use Kerberos .To Implement Hadoop Security By implementing Kerberos which is like issuing password to your Hadoop user to establish their true Identity. One of the main feature of using Kerberos is the password never flows through the network as all the principals are on the local Kerberos server. The Kerberos server is also called the Key Distribution Centre (KDC).The Cluster resources (host or services) uses the keytab file to store their passwords to authenticate itself against the Kerberos without entering a password each time they are accessing the Hadoop Services.

The keytab file generated automatically by Ambari as we configure Kerberos in the cluster using Ambari to work with an existing Active Directory installation. Since we don't have Active directory installed, we are using a local MIT KDC installation in this system and use it as a KDC server.

4.1.1 There are multiple ways to configure Kerberos using Ambari.

- [1] An automated installation and configuration which creates keytabs and principals for all the services
- [2] Manual Kerberos setup to create the principals, generate and distribute the keytabs across the nodes.

As ours is a single node cluster we don't need to distribute Keytab.

4.2 Steps to Install Kerberos and Ranger

- [1] Install MIT KDC in local System
- [2] Create a Kerberos database using kdb5_util tool
- [3] Create a Kerberos admin principal using kadmin.local tool
- [4] Create organizational users using kadmin.local tool
- [5] Enable Kerberos using Ambari automated installation wizard
- [6] Ambari will create key tab files for all the service users and place them in /etc/security/keytabs/ folder of all the hosts
- [7] Configure ranger user-sync to download the organizational users from Kerberos database
- [8] Enable Ranger plugin for Hadoop services like HDFS, Hive
- [9] Create an HDFS policy and ranger policy for providing default access to the users

V. EXPERIMENTAL EVALUATION

Steps	Operation	Result
1. Before login	hdfsdfs-ls/	Failed,Required Login.
2. Login and get ticket	Kinittestuser/testgroup	Ticket generated which can be seen with the help of klist command.
3. After Login	hdfsdfs -ls /	Failed Access denied.
4. Authorization through Apache Ranger	Add user and Generate Policy for the user.	Create access policy in HDFS for the test user with read and execute

		permissions.
5. After Authorization	hdfsdfs -ls /	Gain access for listing files in the hadoop file system.

5.1 Internal User and External User

5.1.1 Internal Users

The internal users are the LINUX System users which Ranger user sync server synchronizes from the local Linux OS. They are not always allowed access to Ranger UI portal.

5.1.2 External Users

The external users (can be from AD/LDAP) are sync by Ranger user sync server to be used for policy creation.

Authentication Protocol	Authentication Scheme	Authorization Scheme	Description
Kerberos	Kerberos-Ticket	Internal	The Client is Authenticated by the Kerberos Authentication Server and issued a TICKET to access the Hadoop file system. The user must exist in the Kerberos database.
Kerberos	Application level	Internal	The user is authenticated to the application server and issued a Kerberos session key.
Apache ranger	Application level	Internal	The user's internal name matches with the Kerberos Principal.

VI. RESULTS AND CONCLUSION

We are using `kadmin.local` utility provided by MIT KDC to create admin user for KDC in Fig.3. Ambari provides a wizard to help with enabling Kerberos in the cluster in Fig.4. Also, we have deployed the Java Cryptography Extension (JCE) security policy files on the Ambari Server and on all hosts in the cluster before we enable Kerberos. Kerberos use this JCE to encrypt or decrypt the Kerberos ticket it generates. In this Adminuser trying to get the initial credentials from KDC Database of Kerberos in Fig.5. Ticket generated by the TGS for admin user in Fig.6. Ranger Creating Policy for user and Groups for Different Files and Directories in Fig.7.

```
[root@sandbox ~]# kadmin.local -q "addprinc admin/admin"
Authenticating as principal root/admin@HORTONWORKS.COM with password.
WARNING: no policy specified for admin/admin@HORTONWORKS.COM; defaulting to no policy
Enter password for principal "admin/admin@HORTONWORKS.COM":
Re-enter password for principal "admin/admin@HORTONWORKS.COM":
Principal "admin/admin@HORTONWORKS.COM" created.
[root@sandbox ~]#
```

Fig.3 Create a Kerberos Admin

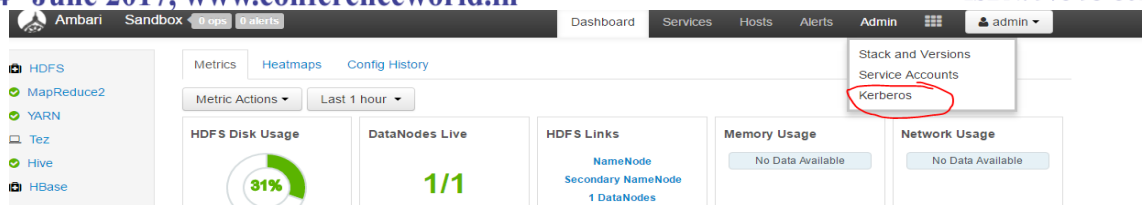


Fig.4 Enable Kerberos Using Ambari

```
[root@sandbox ~]# kinit admin/admin
Password for admin/admin@HORTONWORKS.COM:
```

Fig.5 Admin user trying to get the initial credentials from KDC Database of Kerberos

```
[root@sandbox ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: admin/admin@HORTONWORKS.COM

Valid starting Expires Service principal
04/25/17 06:39:38 04/26/17 06:39:38 krbtgt/HORTONWORKS.COM@HORTONWORKS.COM
renew until_ 04/25/17 06:39:38
```

Fig.6 The ticket generated by the TGS for admin user

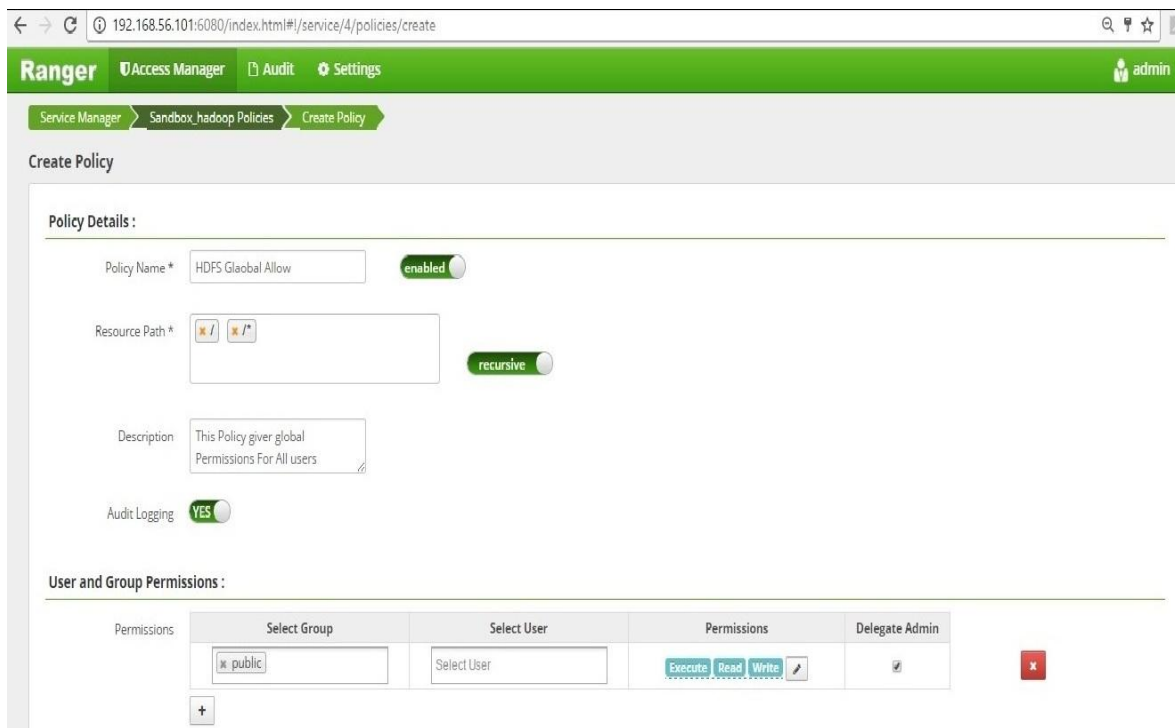


Fig.7 Creating Policy in Apache Ranger

VI. FUTURE SCOPE

In future we can use apache Knox which is centralized gateway for Hadoop cluster validation .It is a single gateway to validates user password for getting entry to the different services in the cluster. Secondly, we can

encrypt Data at rest and Data Encryption over transmission which is encryption of data as it moves we can refer TDE (Transparent Data Encryption) with Ranger KMS (Key management System) which is inbuilt in ranger.

REFERENCES

- [1] Smith, Kevin T. "Big Data Security: The Evolution of Hadoop's Security Model." (2013). <http://www.infoq.com/articles/HadoopSecurityModel>
- [2]"Hadoop Security Analysis" (2013) <http://www.tuicool.com/articles/NFf6be>
- [3]Agrawal,Sumeet Kumar."How to Leverage Big Data Security with Informatica and hortonworks" (2015)<https://hortonworks.com/blog/how-to-leverage-big-data-security-with-informatica-and-hortonworks/>
- [4] Lakhe, Bhushan. "Hadoop Metrics and Their Relevance to Security "Practical Hadoop Security. Apress, 2014.183-189.
- [5] Neuman, Clifford, et al. "The Kerberos network authentication service (V5)." (2005).
- [6] "Apache Ranger Architecture" https://hortonworks.com/apache/ranger/#section_2
- [7]"What Is Apache Hadoop?" (2017)<http://hadoop.apache.org>
- [8] Zheng, Kai, and Weihua Jiang" A token authentication solution for hadoop based on kerberos pre-authentication." Data Science and Advanced Analytics (DSAA), 2014 International Conference on. IEEE,2014.
- [9] Zissis, Dimitrios, and DimitriosLekkas. "Addressing cloud computing security issues" Future Generation computer systems 28.3 (2012): 583-592.
- [10] Wankhede, Paresh, and Nayanjyoti Paul."Secure and multi-tenant Hadoop cluster-an experience"Green High Performance Computing (ICGHPC),2016 2nd International Conference on. IEEE,2016.