Ranger User Guide (work in progress)

- Summary
- Prerequisites
- Opening the console
- Repository Manager
- Repository configuration
    - HDFS Repository configuration
    - Hive Repository configuration
    - HBase Repository configuration
    - Knox Repository configuration
    - Storm Repository configuration
- Policy Manager
- Policy creation
    - HDFS Policy creation

Summary

This document details how to use properly Apache Ranger in manage security in your cluster.

These instructions are for using Ranger on CentOS/RHEL (release 6).

This page is currently being written. Please help us by sending your remarks.

Prerequisites

- Apache Ranger must have been installed on your cluster
- Ranger plugins need to be configured for the services you want to securize
    - If you think you may have missed one or several steps, you can check this Ranger Installation Guide

Opening the console

First things first, in order to access to the Ranger Administrating Console, you have to log in to the Ranger Interface. By default, this interface is available on the Ambari server on the port 6080.

Once you logged in, you can see your username on the top right-hand corner of Ranger Console home page. Clicking on it allows you to log out, by a simple click on **Logout** at the drop-down menu.

The console allows five types of functions :

- The Repository Manager (visible upon user login) : add and manage service repositories
- The Policy Manager tab : create and manage repository policies
- The Users/Groups tab : assign policy permissions to users and groups
- The Analytics tab : perform analytics on one or more HDFS, Hive, HBase, Knox or Storm policies
- The Audit tab : monitor user activity at the resource level, and conditional auditing based on users, group or time

Repository Manager

The Repository Manager is opened by defaul after you log into the Ranger Console. Therefore to access to it from any tab in the Ranger Console, simply click on **Ranger** at the top left corner.

- To add a new repository to the Policy Manager : click the **+** button in the corresponding box on Ranger Policiy Manager, then complete the required information. When the screen have been fulfilled, click the green **Add button**.

- To edit an existing repository : in the Policy Manager, click the **Edit** icon to the right of the entry for that repository. The Policy Manager then displays an expanded view of that repository, including a list of the policies it contains, their current status, and the groups designated to administer those policies.

- To delete a repositoy from the Policy Manager : click the **Delete** icon to the right of the entry for that repository.

Repository configuration

- **HDFS Repository configuration**
    - Repository Name : *name of the repository; required when configuring agents*
    - Description : *a description of the repository*
    - Active status : *Enabled or Disabled*
    - Repository Type : *HDFS (cannot be modified)*
    - User Name : *end system username that can be used for connection*

    - fs.default.name : *location of the Hadoop HDFS service, as noted in the Hadoop configuration file core-site.xml OR (if this is a HA environment) the path for the primary NameNode*
    - hadoop.security.authorization : *true or false, as specified in core-site.xml, to enable authorization for different protocols or not.*
    - hadoop.security.authentication : *type of authentication in use, as noted in the Hadoop configuration file core-site.xml. Can be either simple or Kerberos (required only if authorization is enabled)*
    - hadoop.security.auth_to_local : *maps the login credential to a username with Hadoop. Use the value noted in the Hadoop configuration file, core-site.xml*
    - dfs.datanode.kerberos.principal : *principal associated with the DataNode where the repository resides, as noted in the Hadoop configuration file hdfs-site.xml (required only if Kerberos authentication is enabled)*
    - dfs.namenode.kerberos.principal : *principal associated with the NameNode where the repository resides, as noted in the Hadoop configuration file hdfs-site.xml (required only if Kerberos authentication is enabled)*
    - dfs.secondary.namenode.kerberos.principal : *principal associated with the secondary NameNode where the repository resides, as noted in the Hadoop configuration file hdfs-site.xml (required only if Kerberos authentication is enabled)*
    - hadoop.rpc.protection : *a comma-separated list of protection values for secured SASL connections. Possible values are authentication, integrity and privacy.*
    - Common Name For Certificate : *name of the certificate*

- **Hive Repository configuration**
    - Repository Name : *name of the repository; required when configuring agents*
    - Description : *a description of the repository*
    - Active status : *Enabled or Disabled*
    - Repository Type : *Hive (cannot be modified)*
    - User Name : *end system username that can be used for connection*
    - Password : *password for the username entered above*
    - jdbc.driver ClassName : *the full classname of the driver used for Hive connections. Default is org.apache.hive.jdbc.HiveDriver*
    - jdbc.url : *the complete connection URL, including port and database name. For example, on HortonWorks sandbox : jdbc: hive2://sandbox:10000/ -- Default port is 10000*
    - Common Name For Certificate : *name of the certificate*

- **HBase Repository configuration**
    - Repository Name : *name of the repository; required when configuring agents*
    - Description : *a description of the repository*
    - Active status : *Enabled or Disabled*
    - Repository Type : *HBase (cannot be modified)*
    - User Name : *end system username that can be used for connection*
    - Password : *password for the username entered above*
    - hadoop.security.authorization : *type of authorization in use. Can be either simple or Kerberos.*
    - hbase.master.kerberos.principal : *Kerberos principal for the HBase Master (required only if Kerberos authetication is enabled)*
    - hbase.security.authentication : *As noted in the Hadoop configuration file hbase-site.xml*
    - hbase.zookeeper.property.cientPort : *As noted in the Hadoop configuration file hbase-site.xml*
    - hbase.zookeeper.quorum :*As noted in the Hadoop configuration file hbase-site.xml*
    - zookeeper.znode.parent : *As noted in the Hadoop configuration file hbase-site.xml*

- **Knox Repository configuration**
    - Repository Name : *name of the repository; required when configuring agents*
    - Description : *a description of the repository*
    - Active status : *Enabled or Disabled*
    - Repository Type : *Knox (cannot be modified)*
    - User Name : *end system username that can be used for connection*
    - Password : *password for the username entered above*
    - knox.url : *gateway URL for Knox*
    - Common Name For Certificate : *name of the certificate*

- **Storm Repository configuration**
    - Repository Name : *name of the repository; required when configuring agents*
    - Description : *a description of the repository*
    - Active status : *Enabled or Disabled*
    - Repository Type : *Storm (cannot be modified)*
    - User Name : *end system username that can be used for connection*
    - Password : *password for the username entered above*
    - nimbus.url : *hostname of nimbus format, in the form :* http://**ipaddress**:8080
    - Common Name For Certificate : *name of the certificate*

Policy Manager

To take a closer look to the policies associated with each repository, go to the service where the repository resides and click the **Edit** button. The Rabger Policy Manager view then opens and displays a view of that repository, with the policies listed beneath. For providing a better access to the policies, this view includes a search window.

- To add a new policy : click the **Add New Policy** button. The form may look slightly different, depending on the type of the repository to which your are adding the policy

- To edit a policy : click the **Edit** icon to the right of the entry for that repository. The Policy Manager displays an expanded view of that policy that you may interact with

- To delete a policy : click the **Delete** icon to the right of the entry for that repository

Policy creation

- ### HDFS Policy creation

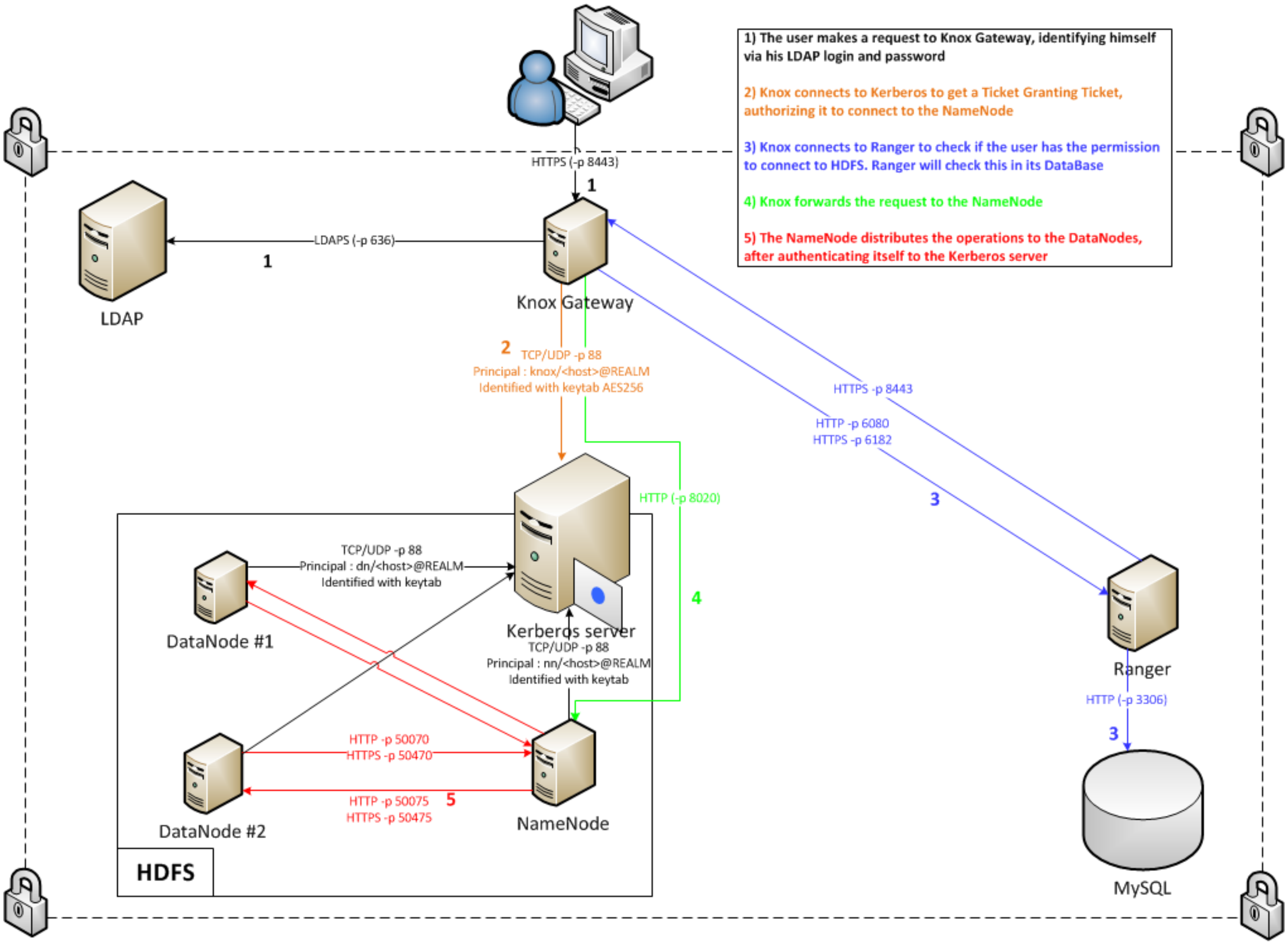
  Through configuration, Apache Ranger enables both Ranger policies and HDFS permission to be checked for a user request. Then, when the NameNode receives a user request, the Ranger Plugin checks for policies set through the Ranger Policy Manager. Then, if there are no policies authorizing the request, the Ranger plugin checks for permissions set in HDFS.

  Thus, for an effective management of the policies via Ranger, we recommand that permissions be created at the Ranger Policy Manager, and to have very restrictive permissions at the HDFS level.

  To add a policy to an HDFS repository : use the HDFS Add Policy form, and complete it as follows :

  - Enter Policy Name : *a unique name for this policy. The name cannot be duplicated anywhere in the system*
  - Resource Path : *the resource path for the policy folder/file. To avoid the need to supply the full path OR to enable the policy for all subfolders or files, you can either complete this path using wild cards (for example, /home*) or specify that the policy should be Recursive (see below)*
  - Description : *(Optional) the purpose of the policy*
  - Recursive : *select if all files or subfolders within the existing folder will be included in this policy. (Use this option if you have specified a specific Resource Path to the top level folder, but want all subfolders or files to be included)*
  - Audit Logging : *whether this policy is audited by Ranger (de-select to disable auditing)*
  - Group Permissions : *use the pick list to assign group permissions appropriate to this policy. If desired, assign the group Administration privileges for the chosen resource. To add users or groups to the list, click the* **+** *button (for further information, see Users)*
  - User Permissions : *use the pick list to assign individual user permissions appropriate to this policy. If desired, designate on or more users as Administrators for the chosen resource*
  - Enable/Disable : *policies are enabled by default. To restrict user/groupe access for a policy, disable the policy*

Then, if you enabled both Knox and Kerberos to secure your cluster, it should work this way :



1) The user makes a request to Knox Gateway, identifying himself via his LDAP login and password

2) Knox connects to Kerberos to get a Ticket Granting Ticket, authorizing it to connect to the NameNode

3) Knox connects to Ranger to check if the user has the permission to connect to HDFS. Ranger will check this in its DataBase

4) Knox forwards the request to the NameNode

5) The NameNode distributes the operations to the DataNodes, after authenticating itself to the Kerberos server