

Ranger Auditing

Date published: 2020-07-28

Date modified: 2021-12-13



Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Audit Overview.....	4
Managing Auditing with Ranger.....	4
View audit details.....	5
Create a read-only Admin user (Auditor).....	8
Configuring Ranger audit properties for Solr.....	9
Configuring Ranger audit properties for HDFS.....	10
Ranger Audit Filters.....	11
Default Ranger audit filters.....	11
Configuring a Ranger audit filter policy.....	14
How to set audit filters in Ranger Admin Web UI.....	17
Filter service access logs from Ranger UI.....	18
Excluding audits for specific users, groups, and roles.....	20
Changing Ranger audit storage location and migrating data.....	21
Configuring Ranger audits to show actual client IP address.....	25

Audit Overview

Apache Ranger provides a centralized framework for collecting access audit history and reporting data, including filtering on various parameters. Ranger enhances audit information obtained from Hadoop components and provides insights through this centralized reporting capability.

Ranger plugins support storing audit data to multiple audit destinations.

Solr

The Solr audit destination is a short term audit destination (with a default TTL of 90 days) managed by Solr which can be configured by a Ranger Admin user. The Ranger Admin Web UI displays the access audit data from the audit data stored in Solr.

HDFS

The HDFS audit destination is a long term audit destination for archival/compliance purposes. The HDFS audit destination has no default retention/purge period. A customer must manage the storage/retention/purge/archival of audit data stored in HDFS manually.

Related Information

[Configuring Ranger audit properties for Solr](#)

[Configuring Ranger audit properties for HDFS](#)

Managing Auditing with Ranger

To explore options for auditing policies in Ranger, click Audit in the top menu.

Exclude Service Users : ☐

Entries : 1 to 25 of 149 | Last Updated Time : 07/21/2019 12:24:11 PM

Policy ID	Policy Version	Event Time	Application	User	Service Name / Type	Resource Name / Type	Access Type	Result	Access Enforcer	Agent Host Name	Client IP	C
3	1	07/21/2019 12:21:35 PM	hbaseMaster	hbase	cm_hbase	--	balance	Allowed	ranger-acl	dhoyle-7-1-1.vpc.cloudera.com		C
3	1	07/21/2019 12:16:30 PM	hbaseMaster	hbase	cm_hbase	--	balance	Allowed	ranger-acl	dhoyle-7-1-1.vpc.cloudera.com		C
3	1	07/21/2019 12:11:30 PM	hbaseMaster	hbase	cm_hbase	--	balance	Allowed	ranger-acl	dhoyle-7-1-1.vpc.cloudera.com		C
3	1	07/21/2019 12:06:30 PM	hbaseMaster	hbase	cm_hbase	--	balance	Allowed	ranger-acl	dhoyle-7-1-1.vpc.cloudera.com		C

There are six tabs on the Audit page:

- Access
- Admin
- Login sessions
- Plugins
- Plugin Status
- User Sync

View audit details

How to view operation details in Ranger audits.

Procedure

To view details for a particular operation, click any tab, then Policy ID, Operation name, or Session ID.

Audit > Access: HBase Table

Ranger

Access Manager

Audit

Security Zone

Settings

admin

Access

Admin

Login Sessions

Plugins

Plugin Status

User Sync

START DATE: 07/21/2019

Exclude Service Users: ☐

Entries: 1 to 25 of

Policy ID	Policy Version	Event Time	Application	User	Service Name / Type	Resource Name / Type	Access Type	Result	Access
3	1	07/21/2019 12:51:30 PM	hbaseMaster	hbase	cm_hbase hbase	--	balance	Allowed	ranger-
3	1	07/21/2019 12:46:30 PM	hbaseMaster	hbase	cm_hbase hbase	--	balance	Allowed	ranger-
3	1	07/21/2019 12:41:30 PM	hbaseMaster	hbase	cm_hbase hbase	--	balance	Allowed	ranger-acl
3	1	07/21/2019 12:36:30 PM	hbaseMaster	hbase	cm_hbase hbase	--	balance	Allowed	ranger-acl dhoyle-7-1-1.vpc.cloudera.com C
3	1	07/21/2019 12:31:31 PM	hbaseMaster	hbase	cm_hbase hbase	--	balance	Allowed	ranger-acl dhoyle-7-1-1.vpc.cloudera.com C
3	1	07/21/2019 12:26:30 PM	hbaseMaster	hbase	cm_hbase hbase	--	balance	Allowed	ranger-acl dhoyle-7-1-1.vpc.cloudera.com C

Policy Details

Service Name: cm_hbaseService Type: hbase

Policy Details:

Policy Type: Access

Policy ID: 3

Version: 1

Policy Name: all - table, column-family, column

HBase Table: Enabled

HBase Column-family: Include

HBase Column: Include

Description: Policy for all - table, column-family, column

Audit Logging: True

Policy Labels: --

Allow Condition: Version 1

OK

Ranger
Access Manager Audit Security Zone Settings
admin

ACCESS
ADMIN
Login Sessions
Plugins
Plugin Status
User Sync

Entries : 1 to 25 of 70 | Last Updated Time : 07/21/2019 01:09:40 PM

Operation	Audit Type	User	Date (Eastern Daylight Time)	Actions	Session Id
Service updated tag_service2	Ranger Service	admin	07/21/2019 01:09:34 PM	Update	40
Group created temp_employees	Ranger Group	admin	07/20/2019 02:15:05 PM	Create	38
Group created audit	Ranger Group	admin	07/18/2019 04:18:42 PM	Create	35
Exported policies	Ranger Policy	admin	07/17/2019 03:06:22 PM	Export json	32
Service updated tag_service1	Ranger Service		07/15/2019 04:11:25 PM	Update	
Policy created EXPIRES_ON	Ranger Policy		07/15/2019 04:11:25 PM	Create	
Service created new_tag	Ranger Service		07/15/2019 04:11:25 PM	Create	

Operation : update

X

Name : tag_service2

Date : 07/21/2019 01:09:34 PM Eastern Daylight Time

Updated By : admin

■ Added ■ Deleted

Service Details :

Fields	Old Value	New Value
Service Description	--	--
Service Name	tag_tag	tag_service2

OK

Audit > Admin: Create

[illegible]

Audit > User Sync: Sync details

The screenshot shows the Ranger Admin console interface. At the top, there's a navigation bar with 'Ranger', 'Access Manager', 'Audit', 'Security Zone', and 'Settings'. The 'Audit' tab is selected, and the 'User Sync' sub-tab is active. Below the navigation bar, there's a search bar with 'START DATE: 07/21/2019'. The main content area displays a table of sync events. The table has columns: User Name, Sync Source, Number Of New Users, Number Of New Groups, Number Of Modified Users, Number Of Modified Groups, Event Time, and Sync Details. A modal window titled 'Sync Details' is open, showing configuration parameters for the sync process.

User Name	Sync Source	Number Of New		Number Of Modified		Event Time	Sync Details
		Users	Groups	Users	Groups		
rangerusersync	Unix	0	0	0	0	07/21/2019 01:22:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:21:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:20:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:19:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:18:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:17:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:16:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:15:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:14:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:13:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:12:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:11:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:10:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:09:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:08:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:07:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:06:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:05:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:04:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:03:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:02:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:01:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:00:47 PM	[Eye Icon]

Sync Details

Name	Value
Unix	nss
File Name	/etc/passwd
Sync time	07/21/2019 10:21:48 AM
Last modified time	12/31/1969 04:00:00 PM
Minimum user id	500
Minimum group id	0
Total number of users synced	35
Total number of groups synced	39

OK

Create a read-only Admin user (Auditor)

Creating a read-only Admin user (Auditor) enables compliance activities because this user can monitor policies and audit events, but cannot make changes.

About this task

When a user with the Auditor role logs in, they see a read-only view of Ranger policies and audit events. An Auditor can search and filter on access audit events, and access and view all tabs under Audit to understand access events. They cannot edit users or groups, export/import policies, or make changes of any kind.

Procedure

1. Select Settings > Users/Groups/Roles.
2. Click Add New User.

3. Complete the **User Detail** section, selecting Auditor as the role:

The screenshot shows the Ranger 'User Create' interface. The 'User Detail' section is active. The 'User Name' field is filled with 'auditor1'. The 'New Password' and 'Password Confirm' fields are masked with dots. The 'First Name' field is filled with 'Audrey'. The 'Last Name' and 'Email Address' fields are empty. The 'Select Role' dropdown menu is open, showing 'Auditor' as the selected option. The 'Group' field is filled with 'audit'. At the bottom, there are 'Save' and 'Cancel' buttons.

4. Click Save.

Configuring Ranger audit properties for Solr

How to change the default time settings that control how long Ranger keeps audit data collected by Solr.

About this task

The Solr audit destination is intended to store short term audit records .You can configure parameters that control how much data collected by Solr that Ranger will store for auditing purposes.

Table 1: Ranger Audit Configuration Parameters for Solr

Parameter Name	Description	Default Setting	Units
ranger.audit.solr.config.ttl	Time To Live for Solr Collection of Ranger Audits	90	days
ranger.audit.solr.config.delete.trigger	Auto Delete Period in seconds for Solr Collection of Ranger Audits for expired documents	1	days (configurable)



Note: "Time To Live for Solr Collection of Ranger Audits" is also known as the Max Retention Days attribute.

Procedure

1. From Cloudera Manager choose Ranger Configuration .
2. In Search, type ranger.audit.solr.config, then press Return.
3. In ranger.audit.solr.config.ttl, set the the number of days to keep audit data.
4. In ranger.audit.solr.config.delete.trigger set the number and units (days, minutes, hours, or seconds) to keep data for expired documents

5. Refresh the configuration:
 - a) Click Refresh Configuration, as prompted.
 - b) In Actions, click Update Solr config-set for Ranger, then confirm.

Configuring Ranger audit properties for HDFS

How to change the settings that control how Ranger writes audit records to HDFS.

About this task

The HDFS audit destination is intended to store long-term audit records.

You can configure whether Ranger stores audit records in HDFS and at which location.

You must purge long term audit records stored in HDFS manually.

Table 2: Ranger Audit Configuration Parameters for HDFS

Parameter Name	Description	Default Setting	Units
ranger_plugin_hdfs_audit_enabled	controls whether Ranger writes audit records to HDFS	true	T/F
ranger_plugin_hdfs_audit_url	location at which you can access audit records written to HDFS	<hdfs.host_name>string ranger/audit	



Note: You can also disable storing ranger audit data to hdfs in each service specifically by setting `xasecure.audit.destination.hdfs=false` in that service.

Procedure

1. From Cloudera Manager choose Ranger Configuration .
2. In Search, type `ranger_plugin`, then press Return.
3. In `ranger_plugin_hdfs_audit_enabled`, check/uncheck RANGER-1 (Service Wide)
4. In `ranger_plugin_hdfs_audit_url` type a valid directory on the hdfs host.
5. Refresh the configuration, using one of the following two options:
 - a) Click Refresh Configuration, as prompted or, if Refresh Configuration does not appear,
 - b) In Actions, click Update Solr config-set for Ranger, then confirm.

What to do next

(Optional)

You may want to delete older logs from HDFS. Cloudera provides no feature to do this. You may accomplish this task manually, using a script.



Note:

The following example script is not supported by Cloudera. It is shown for reference only. You must test this successfully in a test environment before implementing it in a production cluster.

You must specify the audit log directory by replacing the 2nd line `hdfs dfs -ls /<path_to>/<audit_logs>` in the example script.

You may also include the `-skipTrash` option, if you choose, on 7th line in the script.

```
#####
today=`date +%s`
hdfs dfs -ls /<path_to>/<audit_logs> | grep "^d" | while read line ; do
```

```

dir_date=$(echo ${line} | awk '{print $6}')
difference=$(( ( ${today} - $(date -d ${dir_date} +%s) ) / ( 24*60*60 ) ))
filePath=$(echo ${line} | awk '{print $8}')

if [ ${difference} -gt 30 ]; then
    hdfs dfs -rm -r $filePath
fi
done
#####

```

Related Information

[How to do a cleanup of hdfs files older than a certain date using a bash script](#)

Ranger Audit Filters

You can use Ranger audit filters to control the amount of audit log data collected and stored on your cluster.

About Ranger audit filters

Ranger audit filters allow you to control the amount of audit log data for each Ranger service. Audit filters are defined using a JSON string that is added to each service configuration. The audit filter JSON string is a simplified form of the Ranger policy JSON. Audit filters appear as rows in the Audit Filter section of the Edit Service view for each service. The set of audit filter rows defines the audit log policy for the service. For example, the default audit log policy for the Hadoop SQL service appears in Ranger Admin web UI Service Manager Edit Service when you scroll down to Audit Filter. Audit Filter is checked (enabled) by default. In this example, the top row defines an audit filter that causes all instances of "access denied" to appear in audit logs. The lower row defines a filter that causes no metadata operations to appear in audit logs. These two filters comprise the default audit filter policy for Hadoop SQL service.

Figure 1: Default audit filter policy for the Hadoop SQL service

The screenshot shows the Ranger Admin web UI for the 'Edit Service' view. The 'Audit Filter' section is active, showing a table with the following columns: Is Audited, Access Result, Resources, Operations, Permissions, Users, Groups, and Roles. There are two rows in the table. The first row has 'Yes' for Is Audited, 'DENIED' for Access Result, and empty fields for the others. The second row has 'No' for Is Audited, 'Select Value' for Access Result, and 'METADATA OPERATION' for Operations. A '+' button is highlighted in the bottom left corner of the table area. At the bottom of the form, there are buttons for 'Save', 'Cancel', and 'Delete', along with a 'Test Connection' button.

Default Ranger audit filters

Default audit filters for the following Ranger service appear in Edit Services and may be modified as necessary by Ranger Admin users.

HDFS

Figure 2: Default audit filters for HDFS service

Audit Filter:

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles	
Yes	DENIED	-- 	Type Action Name	Add Permissions +	Select User	Select Group	Select Role	
Yes	Select Value	-- 		Add Permissions +	Select User	Select Group	Select Role	
No	Select Value	-- 	 	Add Permissions +		Select Group	Select Role	
No	Select Value	path/User/ocde/hbase/fo	Type Action Name	Add Permissions +		Select Group	Select Role	
No	Select Value	path/User/spark/applicationHistory	Type Action Name	Add Permissions +		Select Group	Select Role	
No	Select Value	path/User/hue	Type Action Name	Add Permissions +		Select Group	Select Role	
No	Select Value	path/hbase	Type Action Name	Add Permissions +		Select Group	Select Role	
No	Select Value	path/User/history	Type Action Name	Add Permissions +		Select Group	Select Role	
No	Select Value	-- 		Add Permissions +	Select User	Select Group	Select Role	

Hbase

Figure 3: Default audit filters for the Hbase service

Audit Filter:

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles	
Yes	DENIED	-- 	Type Action Name	Add Permissions +	Select User	Select Group	Select Role	
No	Select Value	table/*-ROOF*, *META*, *_id_*, hbase:meta, hbase:acl, default, hbase	Type Action Name	Add Permissions +		Select Group	Select Role	
No	Select Value	table:atlas_janus, ATLAS_ENTITY_AUDIT_EVENTS column-family="column:"	Type Action Name	Add Permissions +		Select Group	Select Role	
No	Select Value	-- 		Add Permissions +		Select Group	Select Role	

Hadoop SQL

Figure 4: Default audit filters for the Hadoop SQL service

Audit Filter:

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles	
Yes	DENIED	-- 	Type Action Name	Add Permissions +	Select User	Select Group	Select Role	
No	Select Value	-- 		Add Permissions +	Select User	Select Group	Select Role	

Knox

Figure 5: Default audit filters for the Knox service

Audit Filter:

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles	
Yes	DENIED	-- 	Type Action Name	Add Permissions +	Select User	Select Group	Select Role	
No	Select Value	-- 	Type Action Name	Add Permissions +		Select Group	Select Role	

Solr

Figure 6: Default audit filters for the Solr service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles	
Yes	DENIED	--	Type Action Name	Add Permissions	Select User	Select Group	Select Role	
No	Select Value	--	Type Action Name	Add Permissions	<div><div>hdfs</div><div>hbase</div><div>hive</div><div>mapred</div><div>oozie</div><div>ranger</div><div>spark</div><div>tez</div></div>	Select Group	Select Role	

Kafka

Figure 7: Default audit filters for the Kafka service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles	
Yes	DENIED	--	Type Action Name	Add Permissions	Select User	Select Group	Select Role	
No	Select Value	topic:ATLAS_ENTITIES, ATLAS_HOOK, ATLAS_SPARK_HOOK	<div><div>describe</div><div>publish</div><div>consume</div></div>	Add Permissions	<div><div>atlas</div></div>	Select Group	Select Role	
No	Select Value	topic:ATLAS_HOOK	<div><div>publish</div><div>describe</div></div>	Add Permissions	<div><div>hive</div><div>hbase</div><div>impala</div><div>nifi</div></div>	Select Group	Select Role	
No	Select Value	topic:ATLAS_ENTITIES	<div><div>consume</div><div>describe</div></div>	Add Permissions	<div><div>rangertagsync</div></div>	Select Group	Select Role	
No	Select Value	consumergroup:*	<div><div>consume</div></div>	Add Permissions	<div><div>atlas</div><div>rangertagsync</div></div>	Select Group	Select Role	
No	Select Value	--	Type Action Name	Add Permissions	<div><div>kafka</div></div>	Select Group	Select Role	

Ranger KMS

Figure 8: Default audit filters for the Ranger KMS service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles	
Yes	DENIED	--	Type Action Name	Add Permissions	Select User	Select Group	Select Role	
No	Select Value	--	<div><div>read</div></div>	Add Permissions	<div><div>keyadmin</div></div>	Select Group	Select Role	

Atlas

Figure 9: Default audit filters for the Atlas service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles	
Yes	DENIED	--	Type Action Name	Add Permissions	Select User	Select Group	Select Role	
No	Select Value	--	Type Action Name	Add Permissions	<div><div>atlas</div></div>	Select Group	Select Role	

ADLS

Figure 10: Default audit filters for the ADLS service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles	
Yes	DENIED	--	Type Action Name	Add Permissions	Select User	Select Group	Select Role	
No	Select Value	--	<div><div>get-status</div><div>read</div><div>list</div></div>	<div><div>List</div><div>Read</div></div>	<div><div>hive</div><div>hbase</div><div>hdfs</div></div>	Select Group	Select Role	

Ozone

Figure 11: Default audit filters for the Ozone service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role
No	Select Value	--	Type Action Name	Add Permissions +	iceom	Select Group	Select Role

S3

Figure 12: Default audit filters for the S3 service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role
No	Select Value	--	ice.read	Add Permissions +	ice.hive ice.hbase ice.hdfs ice.yarn	Select Group	Select Role

Tag-based services

Figure 13: Default audit filters for a tag-based service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role



Note:

Default audit filter policies do not exist for Yarn, NiFi, NiFi Registry, Kudu, or schema registry services.

Configuring a Ranger audit filter policy

You can configure an audit filter as you add or edit a resource- or tag-based service.

To configure an audit filter policy:

1. In Ranger Admin web UI Service Manager click Add or Edit for either a resource-, or tag-based service.
2. Scroll down to Audit Filter.
3. Click Audit Filter flag.

You configure a Ranger audit filter policy by adding (+), deleting (X), or modifying each audit filter row for the service.

4. Use the controls in the filter row to edit filter properties. For example, you can configure:

Is Audited: choose Yes or No

to include or not include a filter in the audit logs for a service

Access Result: choose DENIED, ALLOWED, or NOT_DETERMINED

to include that access result in the audit log filter

Resources: Add or Delete a resource item

to include or remove the resource from the audit log filter

Operations: Add or Remove an action name

to include the action/operation in the audit log filter

(click x to remove an existing operation)

Permissions: Add or Remove permissions

a. Click + in Permissions to open the Add dialog.

b. Select/Unselect required permissions.

For example, in HDFS service select read, write, execute, or All permissions.

Users: click Select User to see a list of defined users

to include one or multiple users in the audit log filter

Groups: click Select Group to see a list of defined groups

to include one or multiple groups in the audit log filter

Roles: click Select Role to see a list of defined roles

to include one or multiple roles in the audit log filter

Audit filter details

- When you save the UI selections described in the preceding list, audit filters are defined as a JSON list. Each service references a unique list.
- For example, ranger.plugin.audit.filters for the HDFS service includes:

```
[
  {
    "accessResult": "DENIED",
    "isAudited": true
  },
  {
    "users": [
      "unaudited-user1"
    ],
    "groups": [
      "unaudited-group1"
    ],
    "roles": [
      "unaudited-role1"
    ],
    "isAudited": false
  },
  {
    "actions": [
      "listStatus",
      "getFileinfo"
    ],
    "accessTypes": [
      "execute"
    ],
  },
]
```

```

    "isAudited":false
  },
  {
    "resources":{
      "path":{
        "values":[
          "/audited"
        ],
        "isRecursive":true
      }
    },
    "isAudited":true
  },
  {
    "resources":{
      "path":{
        "values":[
          "/unaudited"
        ],
        "isRecursive":true
      }
    },
    "isAudited":false
  }
]

```

- Each value in the list is an audit filter, which takes the format of a simplified Ranger policy, along with access results fields.
- Audit filters are defined with rules on Ranger policy attributes and access result attributes.
 - Policy attributes: resources, users, groups, roles, accessTypes
 - Access result attributes: isAudited, actions, accessResult
- The following audit filter specifies that accessResult=DENIED will be audited.

The isAudited flag specifies whether or not to audit.

```
{ "accessResult": "DENIED", "isAudited": true }
```

- The following audit filter specifies that “resource => /unaudited” will not be audited.

```
{ "resources": { "path": { "values": [ "/unaudited" ], "isRecursive": true } }, "isAudited": false }
```

- The following audit filter specifies that access to resource database=> sys table=> dump by user “use2” will not be audited.

```
{ "resources": { "database": { "values": [ "sys" ] }, "table": { "values": [ "dump" ] } }, "users": [ "user2" ], "isAudited": false }
```

- The following audit filter specifies that access result in actions => listStatus, getFileInfo and accessType => execute will not be audited.

```
{ "actions": [ "listStatus", "getFileinfo" ], "accessTypes": [ "execute" ], "isAudited": false }
```

- The following audit filter specifies that access by user “superuser1” and group “supergroup1” will not be audited.

```
{ "users": [ "superuser1" ], "groups": [ "supergroup1" ], "isAudited": false }
```

- The following audit filter specifies that access to any resource tagged as NO_AUDIT will not be audited.

```
{ "resources": { "tag": { "values": [ "NO_AUDIT" ] } }, "isAudited": false }
```

How to set audit filters in Ranger Admin Web UI

You can set specific audit filter conditions for each service, using Create/Edit Service .

About this task

Creating audit filters for a service using the Ranger Admin Web UI can prevent audit logs from being sent to destinations like SOLR and HDFS.

Procedure

1. In the Ranger Admin Web UI Service Manager , click Add New Service or Edit (existing service).
2. On Create/Edit Service, scroll down to Audit Filters.
 - a) Verify that Audit Filter is checked.

Optionally, define any of the following to include in the filter definition:

Is Audited

Defines whether audit logs are stored or not.

Is Audited=Yes: stores audit records in the defined audit destination.

Is Audited=No: do not store audit records.

Access Results

Denied, Allowed, or Not Determined

select to filter access=denied, access=allowed or all by selecting access=Not determined.

Resource

use Resource Details to include or exclude specific resources such as databases, tables, or columns.

Operations

select specific operations to filter



Note: For Operations field, you can refer to the Ranger access audit page and find out the available access types.

Permissions

select specific permissions

Users, Groups, Roles

select specific users, groups, and roles

- b) Click Save.

Figure 14: Adding an audit filter that stores user systest, access=Allowed logs for Hive service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles	
Yes	ALLOWED		Type Action Name	Create	systest x	Select...	Select...	X

3. Test your filters to verify that defined audit filters perform as expected.

Results

Defining specific filtering properties can prevent access logs for service users from being stored in the configured audit destination, if Is Audited = No.

Filter service access logs from Ranger UI

You can limit display of system access/audit log records generated by service users in each service.

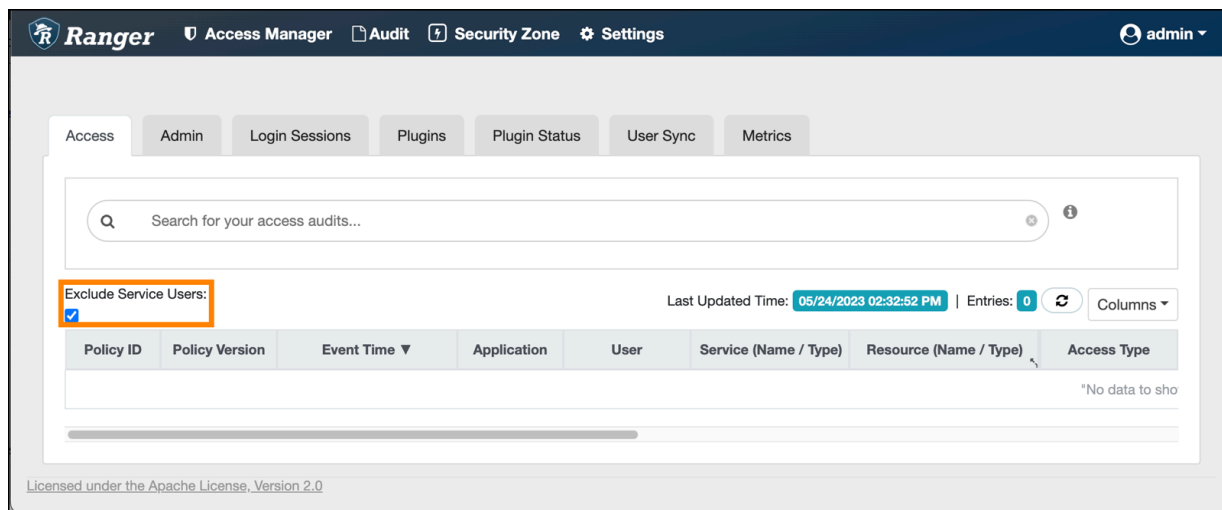
About this task

This topic describes how to limit the display of access log records on the Access tab in the Ranger Admin Web UI.

Procedure

1. Go to Ranger Admin Web UI Audit Access .
2. Check the Exclude Service Users box, as shown in:

Figure 15: Setting the Exclude Service Users flag to true



3. Define specific component services and users for access logs to filter out, in ranger-admin-site.xml.

- a) Go to Cloudera Manager Ranger Configuration
- b) In Search, type ranger-admin-site.
- c) Define the following properties:

Name

ranger.plugins.<service_name>.serviceuser

Value

<service_name>

Name

ranger.accesslogs.exclude.users.list

Value

user1, user2

Figure 16: Filtering out service and user logs for Hive service

Ranger Admin Advanced
Configuration Snippet (Safety
Valve) for conf/ranger-admin-
site.xml

 conf/ranger-admin-
site.xml_role_safety_valve

Ranger Admin Default Group [Undo](#)

[View as XML](#)

Name	<input type="text" value="ranger.accesslogs.exclude.users.list"/>	
Value	<input type="text" value="test1"/>	
Description	<input type="text"/>	
	<input type="checkbox"/> Final	

Name	<input type="text" value="ranger.plugins.hive.serviceuser"/>	
Value	<input type="text" value="hive"/>	
Description	<input type="text"/>	
	<input type="checkbox"/> Final	

4. Click Save Changes (CTRL+S).

5. Restart the Ranger service.

Results

Setting Exclude Service Users to true and defining specific filtering properties prevents audit logs from service users from appearing on Ranger Admin Web UI Audit Access , but does NOT prevent access logs for service users from being generated in Solr.

Excluding audits for specific users, groups, and roles

You can exclude audit records for specific users, groups, and roles from each service from appearing in the Ranger UI.

About this task

Ranger default log functionality creates audit log records for access and authorization requests, specifically around service accounts such as hbase, atlas and solr. Writing so much data to solr can limit the availability of Solr for further usage. This topic describes how to exclude audit records for specific users, groups, and roles from each service from appearing in the Ranger UI. Excluding specific users, groups or roles is also known as creating a blacklist for Ranger audits.

Procedure

1. In the Ranger Admin Web UI Service Manager , click Add New Service or Edit (existing service).
2. On Create/Edit Service, scroll down to Config Properties Add New Configurations .
3. Remove all audit filters from the existing service.

4. Click +, then type one of the following property names:

- ranger.plugin.audit.exclude.users
- ranger.plugin.audit.exclude.groups
- ranger.plugin.audit.exclude.roles

followed by one or more values.



Note: You can include multiple values for each exclude property using a comma-separated list.

Figure 17: Adding an exclude users property to the HadoopSQL service

Name	Value	
tag.download.auth.users	hive,hdfs,impala	✕
policy.download.auth.users	hive,hdfs,impala	✕
policy.grantrevoke.auth.users	hive,impala	✕
enable.hive.metastore.lookup	true	✕
default.policy.users	impala,hive,hue,beacon,admin,dpt	✕
hive.site.file.path	/etc/hive/conf/hive-site.xml	✕
ranger.plugin.audit.exclude.users	testuser2	✕

Below the table is a '+' button to add new configurations.

After adding the above configuration; if testuser2 user performs any actions for HadoopSQL service, Audit Access logs will not appear in the Ranger UI, but are still sent to Solr.

Similarly, you can exclude (or blacklist) users belonging to a particular group or role by adding a user-specific or role-specific configuration.

Changing Ranger audit storage location and migrating data

How to change the location of existing and future Ranger audit data collected by Solr from HDFS to a local file system or from a local file system to HDFS.

Before you begin

- Stop Atlas from Cloudera Manager.

- If using Kerberos, set the SOLR_PROCESS_DIR environment variable.

```
# export SOLR_PROCESS_DIR=$(ls -ldtr /var/run/cloudera-scm-agent/process/
*SOLR_SERVER | tail -1)
```

About this task

Starting with Cloudera Runtime version 7.1.4 / 7.2.2, the storage location for ranger audit data collected by Solr changed to local file system from HDFS, as was true for previous versions. The default storage location Ranger audit data storage location for Cloudera Runtime-7.1.4+ and Cloudera Runtime-7.2.2+ installations is local file system. After upgrading from an earlier Cloudera platform version, follow these steps to backup and migrate your Ranger audit data and change the location where Solr stores your future Ranger audit records.

- The default value of the index storage in the local file system is /var/lib/solr-infra. You can configure this, using Cloudera Manager Solr Configuration "Solr Data Directory" .
- The default value of the index storage in HDFS is /solr-infra. You can configure this, using Cloudera Manager Solr Configuration "HDFS Data Directory" .

Procedure

1. Create HDFS Directory to store the collection backups.

As an HDFS super user, run the following commands to create the backup directory:

```
# hdfs dfs -mkdir /solr-backups
# hdfs dfs -chown solr:solr /solr-backups
```

2. Obtain valid kerberos ticket for Solr user.

```
# kinit -kt solr.keytab solr/$(hostname -f)
```

3. Download the configs for the collection.

```
# solrctl instancedir --get ranger_audits /tmp/ranger_audits
# solrctl instancedir --get atlas_configs /tmp/atlas_configs
```

4. Modify the solrconfig.xml for each of the configs for which data needs to be stored in HDFS.

In /tmp/<config_name>/conf created during Step 3., edit properties in the solrconfig.xml file as follows:

- When migrating your data storage location from a local file system to HDFS, replace these two lines:

```
<directoryFactory name="DirectoryFactory"
  class="${solr.directoryFactory:solr.NRTCachingDirectoryFactory}">
<lockType>${solr.lock.type:native}</lockType>
```

with

```
<directoryFactory name="DirectoryFactory"
  class="${solr.directoryFactory:org.apache.solr.core.HdfsDirectoryFactory}">
<lockType>${solr.lock.type:hdfs}</lockType>
```

- When migrating your data storage location from HDFS to a local file system, replace these two lines:

```
<directoryFactory name="DirectoryFactory"
  class="${solr.directoryFactory:org.apache.solr.core.HdfsDirectoryFactory}">
<lockType>${solr.lock.type:hdfs}</lockType>
```

with

```
<directoryFactory name="DirectoryFactory"
  class="${solr.directoryFactory:solr.NRTCachingDirectoryFactory}">
<lockType>${solr.lock.type:native}</lockType>
```

5. Backup the Solr collections.

- When migrating your data storage location from a local file system to HDFS, run:

```
# curl -k --negotiate -u : "https://$(hostname
-f):8995/solr/admin/collections?action=BACKUP&name=vertex_backup&col
lection=vertex_index&
location=hdfs://<Namenode_Hostname>:8020/solr-backups"
```

In the preceding command, the important points are name, collection, and location:

name

specifies the name of the backup. It should be unique per collection

collection

specifies the collection name for which the backup will be performed

location

specifies the HDFS path, where the backup will be stored

Repeat the curl command for different collections, modifying the parameters as necessary for each collection.

The expected output would be -

```
"responseHeader": {
  "status": 0,
  "QTime": 10567},
"success": {
  "Solr_Server_Hostname:8995_solr": {
    "responseHeader": {
      "status": 0,
      "QTime": 8959}}}}
```

- When migrating your data storage location from HDFS to a local file system:

Refer to Back up a Solr collection for specific steps, and make the following adjustments:

- If TLS is enabled for the Solr service, specify the trust store and password by using the ZKCLI_JVM_FLAGS environment variable before you begin the procedure.

```
# export ZKCLI_JVM_FLAGS="-Djavax.net.ssl.trustStore=/path/to/
truststore.jks -Djavax.net.ssl.trustStorePassword="
```

- Create Snapshot

```
# solrctl --jaas $SOLR_PROCESS_DIR/jaas.conf collection --create-
snapshot <snapshot_name> -c <collection_name>
```

- or use the Solr API to take the backup:

```
curl -i -k --negotiate -u : "https://(hostname -f):8995/solr/admin/
collections?
action=BACKUP&name=ranger_audits_bkp&collection=ranger_audits&location=/
path/to/solr-backups"
```

- Export Snapshot

```
# solrctl --jaas $SOLR_PROCESS_DIR/jaas.conf collection
--export-snapshot <snapshot_name> -c <collection_name> -d
<destination_directory>
```



Note: The <destination_directory> is a HDFS path. The ownership of this directory should be solr:solr.

6. Update the modified configs in Zookeeper.

```
# solrctl --jaas $SOLR_PROCESS_DIR/jaas.conf instancedir --update
  atlas_configs /tmp/atlas_configs
# solrctl --jaas $SOLR_PROCESS_DIR/jaas.conf instancedir --update
  ranger_audits /tmp/ranger_audits
```

7. Delete the collections from the original location.

All instances of Solr service should be up, running, and healthy before deleting the collections. Use Cloudera Manager to check for any alerts or warnings for any of the instances. If alerts or warnings exist, fix those before deleting the collection.

```
# solrctl collection --delete edge_index
# solrctl collection --delete vertex_index
# solrctl collection --delete fulltext_index
# solrctl collection --delete ranger_audits
```

8. Verify that the collections are deleted from the original location.

```
# solrctl collection --list
```

This will give an empty result.

9. Verify that no leftover directories for any of the collections have been deleted.

- When migrating your data storage location from a local file system to HDFS:

```
# cd /var/lib/solr-infra
```

Get the value of "Solr Data Directory, using Cloudera Manager Solr Configuration .

```
# ls -ltr
```

- When migrating your data storage location from HDFS to a local file system, replace these two lines:

```
# hdfs dfs -ls /solr/<collection_name>
```



Note: If any directory name which starts with the collection name deleted in Step 7. exists, delete/ move the directory to another path.

10. Restore the collection from backup to the new location.

Refer to Restore a Solr collection, for more specific steps.

```
# curl -k --negotiate -u : "https://$(hostname
-f):8995/solr/admin/collections?
action=RESTORE&name=<Name_of_backup>&location=hdfs:/
<<Namenode_Hostname>:8020/solr-backups&collection=<Collection_Name>"
```

```
# solrctl collection --restore ranger_audits
-l hdfs://<Namenode_Hostname>:8020/solr-backups
-b ranger_backup -i ranger1
```

The request id must be unique for each restore operation, as well as for each retry.

To check the status of restore operation:

```
# solrctl collection --request-status <requestId>
```



Note: If the Atlas Collections (vertex_index, fulltext_index and edge_index) restore operations fail, restart the solr service and rerun the restore command. Now, the restart operations should complete successfully.

11. Verify the Atlas & Ranger functionality.

Verify that both Atlas and Ranger audits functions properly, and that you can see the latest audits in Ranger Web UI and latest lineage in Atlas.

- To verify Atlas audits, create a test table in Hive, and then query the collections to see if you are able to view the data.
- You can also query the collections every 20-30 seconds (depending on how other services utilize Atlas/Ranger), and verify if the "numDocs" value increases at every query.

```
# curl -k --negotiate -u : "https://$(hostname -f):8995/solr/edge_index/
select?q=%3A*&wt=json&ident=true&rows=0"
# curl -k --negotiate -u : "https://$(hostname -f):8995/solr/vertex_index/
select?q=%3A*&wt=json&ident=true&rows=0"
# curl -k --negotiate -u : "https://$(hostname -f):8995/solr/
fulltext_index/select?q=%3A*&wt=json&ident=true&rows=0"
# curl -k --negotiate -u : "https://$(hostname -f):8995/solr/
ranger_audits/select?q=%3A*&wt=json&ident=true&rows=0"
```

Configuring Ranger audits to show actual client IP address

How to forward the actual client IP address to audit logs generated from a Ranger plugin.

About this task

Ranger audit logs record the IP address through which Ranger policies grant/authorize access. When Ranger is set up behind a Knox proxy server, the proxy server IP address appears in the audit logs generated for each Ranger plugin. You can configure each plugin to forward the actual client IP address on which that service runs, so that the audit logs for that service more specifically reflect access/authorization activity. You must configure each plugin individually. This topic uses the Hive (Hadoop SQL) service as an example.

Procedure

- From Cloudera Manager choose <service_name> Configuration .
- In <service_name> Configuration Search , type ranger-plugin, then press Return.
- In Ranger Plugin Use X-Forwarded for IP Address, check the box.
- In Ranger Plugin Trusted Proxy IP Address, type the IP address of the Knox proxy server host.

The screenshot shows the Cloudera Manager interface for the HIVE-1 service configuration. The 'Ranger Plugin Use X-Forwarded for IP Address' checkbox is checked. The 'Ranger Plugin Trusted Proxy IP Address' field is set to 'KnoxServerHost.IP.address'. The 'Ranger Plugin URL Auth Filesystem Schemes' field is set to 'hdfs;file;wasb;adl'.

Results

Hive audit logs will now show the IP address of the host on which Hive service runs.