# Apache Sentry概述

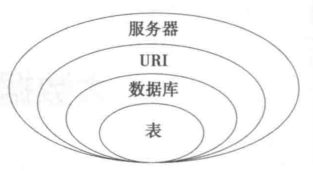
**Apache Sentry是一个用于Hadoop生态系统的细粒度授权系统，与2012年由Cloudera公司进行开发，2016年成为Apache基金的顶级项目。**

## Sentry技术概况

Apache Sentry和Apache Ranger类似，对Hadoop生态系统上的组件，如Hive、Impala等进行细粒度的数据访问控制，对Hadoop集群上进行身份验证的用户和应用程序的数据提供控制和实施精确的权限控制的功能，在Hadoop生态系统现有的组映射环境下，只需通过对Sentry独有的角色（Role）进行操作，即可轻松实现权限的管理。

和Hadoop原生文件权限控制系统相比，Sentry的功能特性有一些特别之处，下面对一些关键概念进行分析。

1. 对象（Object）：Hadoop原生权限控制在文件层次，而通过Sentry授权规则保护的对象则具有更丰富的层次，例如服务器、数据库、表等，其层次关系如图所示。



1. 验证（Authentication）：如果Hadoop的所有配置都是默认的话，Hadoop不会验证用户。如果用户发起一个请求，声称自己是超级用户，Hadoop会允许用户做任何超级用户可以做的事。而Sentry依赖底层认证系统，如Kerberos或LDAP，通过验证凭据可靠的识别用户。
2. 授权（Authorization）：限制用户对给定资源的访问，在Hadoop中授权的实体为用户，而Sentry是基于角色进行访问控制的，授权的实体是角色，这是一种强大的机制，适用于管理大量用户和数据对象的授权。
3. 用户（User）：在Hadoop和Sentry中均是由系统识别个人的，区别在于上面介绍的验证方式不同。
4. 组（Group）：由认证系统维护的一组用户，Sentry使用在Hadoop中配置的组映射机制，以确保Sentry看到与Hadoop生态系统的其他组件相同的组映射。
5. 特权（Privilege）：允许访问对象的指令或规则，在Hadoop中对于文件具有读、写两级特权，在Sentry中，不同对象可以拥有的特权级别略有不同。
6. 角色（Role）：Sentry独有的特征，为一组特权的集合，或者理解为组合多个访问规则的模板。
7. 授权模型（Authorization models）：定义要接受授权规则的对象和允许的操作粒度，Sentry比Hadoop能够实现更细粒度的授权模型，例如，在SQL模型中，对象可以是数据库或表，操作是选择、插入、创建等。对于搜索模型，对象可以是索引、集合和文档，操作可以为查询、更新等。

与Ranger类似，Sentry也使用了插件-服务器的结构进行权限管理，Sentry通过挂钩（HOOK）将插件嵌入需要数据保护的单个Hadoop组件中，拦截操作请求并进行权限验证，为了实现Sentry可扩展的设计目标，它使用了结合层（Bingding）来标准化授权请求，使得授权认证可以面向多种数据引擎。

## Sentry技术优势

1. 出色的规范性

Sentry在开发过程中考虑到很多国际法案，包括SOX、PCI、HIPAA、EAL3等。

1. 安全的授权机制

在分布式系统中，Sentry通常与Kerberos认证结合使用，Kerberos认证定义允许哪些主机连接到服务器上，Sentry对已通过验证的用户提供数据访问权限，使用Sentry和Kerberos的组合可防止恶意用户通过在不受信任的计算机上创建命名账户来获取服务器上的敏感数据。管理员也可以通过Sentry和带选择语句的视图或UDF，根据需要在文件内屏蔽数据。

1. 细粒度的访问控制

Sentry支持细粒度的Hadoop数据和元数据访问控制。在Hive和Impala中Sentry的发行版本中，Sentry在服务器、数据库、表和视图范围提供了不同特权级别的访问控制，包括查找、插入等，允许管理员使用视图限制对行或列的访问。管理员也可以通过Sentry和带选择语句的视图或UDF，根据需要在文件内屏蔽数据。

1. 基于角色的管理

Sentry存在一条清晰的映射关系，权限🡪角色🡪用户组🡪用户，从权限到角色，从角色到用户组都是通过grant/revoke的SQL语句来实现的，而从用户组到用户则通过Hadoop自身的用户/组映射自动授予。

1. 多租户管理

Sentry允许为委派给不同管理员的不同数据集设置权限。在Hive/Impala的情况下，Sentry可以在数据库/schema级别进行权限管理。

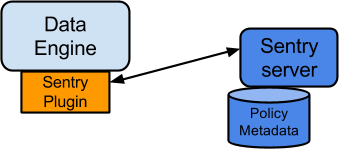
1. 统一平台

Sentry为确保数据安全，提供了一个统一平台，使用现有的Hadoop Kerberos实现安全认证。同时，通过Hive或Impala访问数据时可以使用同样的Sentry协议。

## Sentry架构

### Sentry架构图

Sentry主要由下图所示的3个组件组成，存在于集成配置中。



**Sentry服务器（Sentry Server）**：Sentry服务器管理授权元数据，它支持安全检索和操作元数据的接口。

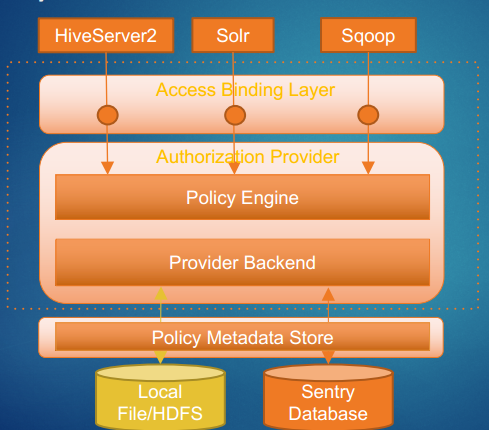
**数据引擎（Data Engine）：**这是需要授权访问数据或元数据的数据处理程序，例如Hive、Impala。数据引擎加载Sentry插件，并且拦截所有访问资源的客户端请求，然后由Sentry插件进行验证。

**Sentry插件（Sentry Plugin）**：Sentry插件在数据引擎中运行，它提供了操作存储在Sentry服务器中的授权元数据的接口，并且包括使用从服务器检索的授权元数据评估访问请求的授权策略引擎。

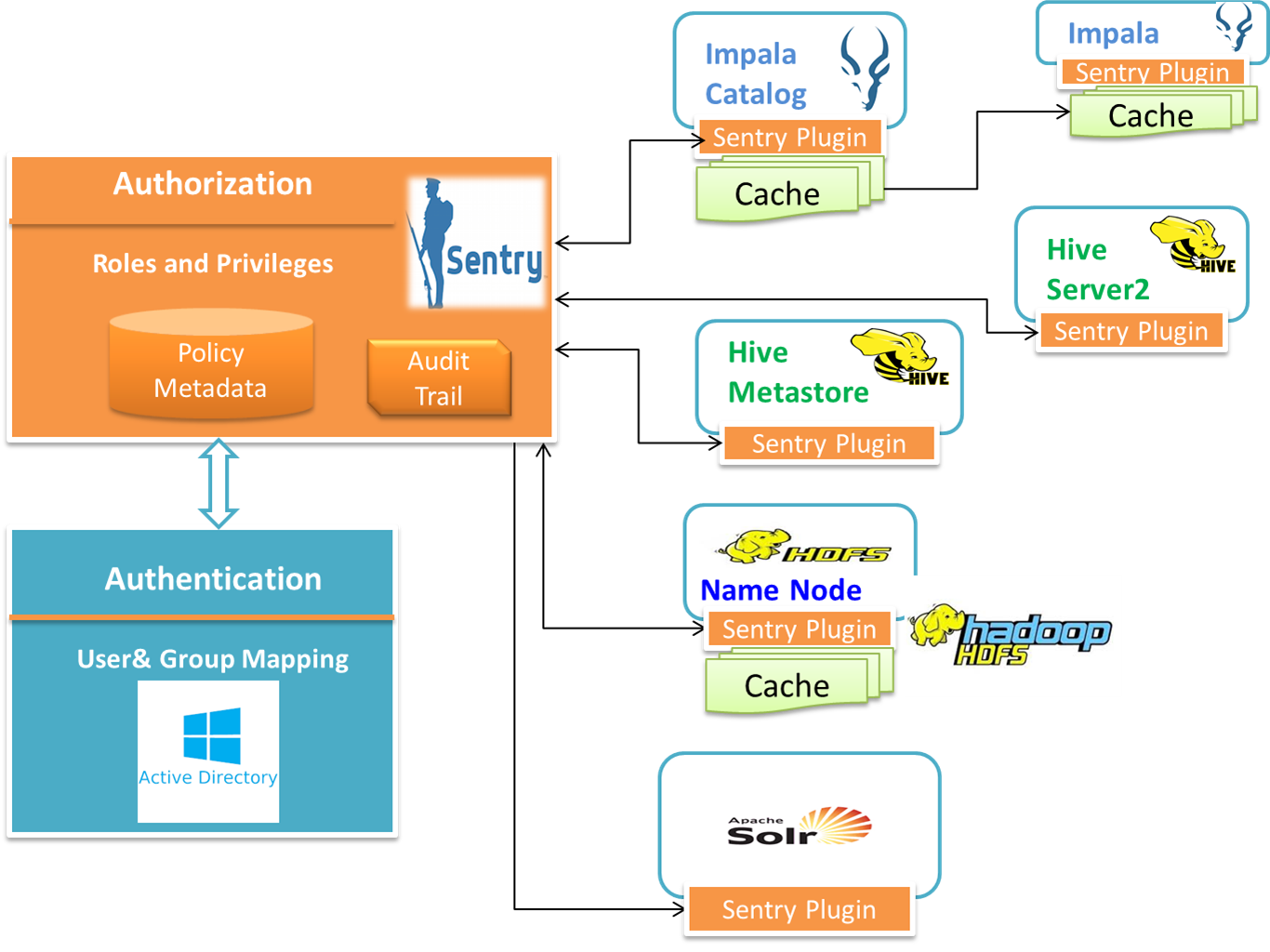
**Policy Metadata**：这里的Metadata存储权限策略数据，对应的会需要一个外部存储db。

实际上，Sentry服务器的主要目的只是管理元数据，真正的授权决策由Sentry插件中的策略引擎做出。

从另一个层秒角度来看Apache Sentry的内部架构：



### Sentry与Hadoop生态圈组件的集成



上图中在HDFS里面多了一个cache层，主要是为了保持HDFS的权限与HIve的一致，NameNode的Sentry Plugin程序会定期拉取Hive的Metadata信息以及Sentry Server上的权限信息，并cache起来，提高性能。

另外地在Sentry Sever中，它还有audit模块，记录了所有模块的请求访问记录。

Sentry架构中包含3个重要的层次：

1. 连接层

Sentry的策略引擎作为Sentry插件的一部分，由Hive等数据引擎调用，而连接层则是数据引擎和Sentry授权之间的桥梁，负责以请求者原生格式获取授权请求，并将其转换为可由Sentry策略引擎处理的授权请求。

与数据引擎集成时，Sentry将自己的HOOK函数插入各SQL引擎的编译、执行不同阶段，HOOK函数收集SQL语句执行的对象，操作等信息。同时连接层将这些信息转换为Sentry授权请求，并将其传递给策略引擎层。

1. 策略引擎层

这是Sentry授权的核心，策略引擎从连接层获取请求的特权，并从策略提供者层获取所需的特权，他比较请求的权限和所需的权限，并决定是否允许操作。

1. 策略提供者层

这是使授权元数据可用于策略引擎层的抽象，这允许元数据的使用与元数据的存储方式无关。目前Sentry支持基于文件的存储和基于关系型数据库的存储。

基于文件的方案是将元数据存储在ini格式的文件中，该文件可以存储在本地文件系统或者HDFS中，文件内包含了组与角色，角色与特权间的两组映射。

# Apache Sentry部署

## 集群情况

三台hadoop集群，分别是master、slave1和slave2。下面是这三台机器的软件分布：

**master**：NameNode、ZK、HiveMetaSotre、HiveServer2、SentryServer

**slave1**：DataNode、ZK

**slave2**：DataNode、ZK

## 软件需求

1. MySql
2. mysql-jdbc.jar：mysql-connector-java-5.1.38-bin.jar
3. Hadoop
4. Hive1.1.0：apache-hive-1.1.0-bin.tar.gz
5. Sentry1.6.0：apache-sentry-1.6.0-incubating-bin.tar.gz   
   注意：这里的Sentry1.6.0版本对应的是Hive1.1.0版本，目前还不支持Hive1.2及以上版本。

## 软件安装

### MySql安装

**在线安装：sudo apt-get install mysql-server。**

**修改文件：vim /etc/mysql/my.cnf，内容如下：**

#bind-address = 127.0.0.1

bind-address = 0.0.0.0

重启mysql：service mysql restart

### Hadoop安装

此处省略

### Hive安装

此处省略

创建hive的元数据mysql数据库

create database hive;

CREATE USER hive IDENTIFIED BY 'hive';

GRANT all ON hive.\* TO hive@'%' IDENTIFIED BY'hive';

flush privileges;

### Sentry安装

**1. 解压安装**

tar -zxvf apache-sentry-1.6.0-incubating-bin.tar.gz -C /opt

**2. 修改sentry的配置文件(sentry-site.xml)**

mv apache-sentry-1.6.0-incubating-bin apache-sentry-1.6.0

cp mysql-connector-java-5.1.38-bin.jar /opt/apache-sentry-1.6.0/lib

cd /opt/apache-sentry-1.6.0/conf

vim sentry-site.xml

**3. 修改sentry-sitem.xml文件**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83 | <?xml version="1.0" encoding="utf-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>sentry.service.admin.group</name>  <value>admin</value>  </property>  <property>  <name>sentry.service.allow.connect</name>  <value>hive,admin</value>  </property>  <property>  <name>sentry.service.reporting</name>  <value>JMX</value>  </property>  <property>  <name>sentry.service.server.rpc-address</name>  <value>192.168.70.110</value>  </property>  <property>  <name>sentry.service.server.rpc-port</name>  <value>8038</value>  </property>  <property>  <name>sentry.store.group.mapping</name>  <value>org.apache.sentry.provider.common.HadoopGroupMappingService</value>  </property>  <property>  <name>sentry.hive.server</name>  <value>server1</value>  </property>  <!-- 配置Webserver -->  <property>  <name>sentry.service.web.enable</name>  <value>true</value>  </property>  <property>  <name>sentry.service.web.port</name>  <value>51000</value>  </property>  <property>  <name>sentry.service.web.authentication.type</name>  <value>NONE</value>  </property>  <property>  <name>sentry.service.web.authentication.kerberos.principal</name>  <value> </value>  </property>  <property>  <name>sentry.service.web.authentication.kerberos.keytab</name>  <value> </value>  </property>  <!--配置认证-->  <property>  <name>sentry.service.security.mode</name>  <value>none</value>  </property>  <property>  <name>sentry.service.server.principal</name>  <value> </value>  </property>  <property>  <name>sentry.service.server.keytab</name>  <value> </value>  </property>  <!-- 配置Jdbc -->  <property>  <name>sentry.store.jdbc.url</name>  <value>jdbc:mysql://192.168.70.110:3306/sentry</value>  </property>  <property>  <name>sentry.store.jdbc.driver</name>  <value>com.mysql.jdbc.Driver</value>  </property>  <property>  <name>sentry.store.jdbc.user</name>  <value>sentry</value>  </property>  <property>  <name>sentry.store.jdbc.password</name>  <value>sentry</value>  </property>  </configuration> |

**4. 配置环境变量**   
编辑vim /etc/profile

export SENTRY\_HOME=/opt/apache-sentry-1.6.0

export PATH=${SENTRY\_HOME}/bin:$PATH

使生效source /etc/profile   
**5. 在mysql中创建sentry数据库**

create database sentry;

CREATE USER sentry IDENTIFIED BY 'sentry';

GRANT all ON sentry.\* TO sentry@'%' IDENTIFIED BY 'sentry';

flush privileges;

**6. 将mysql-jdbc的jar包拷贝到sentry/lib目录下**

cp mysql-connector-java-5.1.38-bin.jar  ${SENTRY\_HOME}/lib

**7. 替换hadoop中jline的jar包**

rm ${HADOOP\_HOME}/share/hadoop/yarn/lib/jline-0.9.94.jar

cp ${SENTRY\_HOME}/lib/jline-2.12.jar ${HADOOP\_HOME}/share/hadoop/yarn/lib

**8. 初始化sentry数据库**

sentry --command schema-tool --conffile  ${SENTRY\_HOME}/conf/sentry-site.xml --dbType mysql --initSchema

**9. 启动sentry服务**

sentry --command service --conffile  ${SENTRY\_HOME}/conf/sentry-site.xml

## Hive集成Sentry

**1. 修改hive-site.xml配置文件**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48 | <?xml version="1.0" encoding="utf-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>hive.sentry.conf.url</name>  <value>file:///opt/apache-hive-1.1.0/conf/sentry-site.xml</value>  </property>  <!-- 配置开关控制列访问权限 -->  <property>  <name>hive.stats.collect.scancols</name>  <value>true</value>  </property>  <!-- Hive Metastore集成Sentry -->  <property>  <name>hive.metastore.pre.event.listeners</name>  <value>org.apache.sentry.binding.metastore.MetastoreAuthzBinding</value>  </property>  <property>  <name>hive.metastore.event.listeners</name>  <value>org.apache.sentry.binding.metastore.SentryMetastorePostEventListener</value>  </property>  <!-- Hive Server2集成Sentry -->  <property>  <name>hive.server2.session.hook</name>  <value>org.apache.sentry.binding.hive.HiveAuthzBindingSessionHook</value>  </property>  <property>  <name>hive.security.authorization.task.factory</name>  <value>org.apache.sentry.binding.hive.SentryHiveAuthorizationTaskFactoryImpl</value>  </property>  <!-- 设置mysql-jdbc -->  <property>  <name>javax.jdo.option.ConnectionURL</name>  <value>jdbc:mysql://192.168.70.110:3306/hive?createDatabaseIfNotExist=true</value>  </property>  <property>  <name>javax.jdo.option.ConnectionDriverName</name>  <value>com.mysql.jdbc.Driver</value>  </property>  <property>  <name>javax.jdo.option.ConnectionUserName</name>  <value>hive</value>  </property>  <property>  <name>javax.jdo.option.ConnectionPassword</name>  <value>hive</value>  </property>  </configuration> |

**2. 修改sentry-site.xml配置文件**   
*注意：该文件是在hive/conf目录下的，不要和sentry/conf目录下的sentry-site.xml文件混淆！*

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | <?xml version="1.0" encoding="utf-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>sentry.service.client.server.rpc-address</name>  <value>192.168.70.110</value>  </property>  <property>  <name>sentry.service.client.server.rpc-port</name>  <value>8038</value>  </property>  <property>  <name>sentry.service.client.server.rpc-connection-timeout</name>  <value>200000</value>  </property>  <!--配置认证-->  <property>  <name>sentry.service.security.mode</name>  <value>none</value>  </property>  <property>  <name>sentry.service.server.principal</name>  <value> </value>  </property>  <property>  <name>sentry.service.server.keytab</name>  <value> </value>  </property>  <property>  <name>sentry.provider</name>  <value>org.apache.sentry.provider.file.HadoopGroupResourceAuthorizationProvider</value>  </property>  <property>  <name>sentry.hive.provider.backend</name>  <value>org.apache.sentry.provider.db.SimpleDBProviderBackend</value>  </property>  <property>  <name>sentry.metastore.service.users</name>  <value>hive</value>  <!--queries made by hive user (beeline) skip meta store check-->  </property>  <property>  <name>sentry.hive.server</name>  <value>server1</value>  </property>  <property>  <name>sentry.hive.testing.mode</name>  <value>true</value>  </property>  </configuration> |

**3. 拷贝sentry/lib下的jar包到hive/lib目录下**

cp ${SENTRY\_HOME}/lib/sentry\*.jar ${HIVE\_HOME}/lib

cp ${SENTRY\_HOME}/lib/shiro-\*.jar ${HIVE\_HOME}/lib

*注意：shiro-\*.jar一定也要拷贝过去，否则会报错。*   
**4. 修改Hive在HDFS上的文件权限**

hadoop fs -chown -R hive:hive /user/hive/warehouse

hadoop fs -chmod -R 770 /user/hive/warehouse

由于HDFS上文件是hive：hive权限，所以linux用户上能   
**5. 启动Hive Metastore服务**

hive --service metastore -hiveconf hive.root.logger=DEBUG,console

**6. 启动Hive Server2服务**

hive --service hiveserver2 -hiveconf hive.root.logger=DEBUG,console

## Hive权限验证

### 准备测试数据

* 创建测试数据文件：vim /tmp/events.csv，内容如下：

10.1.2.3,US,android,createNote

10.200.88.99,FR,windows,updateNote

10.1.2.3,US,android,updateNote

10.200.88.77,FR,ios,createNote

10.1.4.5,US,windows,updateTag

* 然后，在hive中运行下面sql语句（注意：以hive用户启动hive命令）：

create database sensitive;

create table sensitive.events (ip STRING, country STRING, client STRING, action STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

load data local inpath '/tmp/events.csv' overwrite into table sensitive.events;

create database filtered;

create view filtered.events as select country, client, action from sensitive.events;

create view filtered.events\_usonly as select \* from filtered.events where country = 'US';

* 在master上通过 beeline 连接 hiveserver2，运行下面命令创建角色和组：

beeline -u "jdbc:hive2://master:10000/" -n admin -p admin -d org.apache.hive.jdbc.HiveDriver

* 在beelin中执行下面的 sql 语句创建 role、group等：

create role admin\_role;

GRANT ALL ON SERVER server1 TO ROLE admin\_role;

GRANT ROLE admin\_role TO GROUP admin;

create role test\_role;

GRANT ALL ON DATABASE filtered TO ROLE test\_role;

use sensitive;

GRANT SELECT(ip) on TABLE sensitive.events TO ROLE test\_role;

GRANT ROLE test\_role TO GROUP test;

上面创建了两个角色：

* admin\_role，具有管理员权限，可以读写所有数据库，并授权给 admin 和 hive 组（对应操作系统上的组）
* test\_role，只能读写 filtered 数据库，和只能读取sensitive库中events 表中的ip字段，并授权给 test 组。

因为系统上没有test用户和组，所以需要手动创建：useradd test

### 测试

#### 测试admin\_role角色

使用admin用户访问beelin：

beeline -u "jdbc:hive2://master:10000/" -n admin -p admin -d org.apache.hive.jdbc.HiveDriver

admin用户属于admin\_role组，具有管理员权限，可以查看所有角色：

0: jdbc:hive2://master:10000/> show roles;

+-------------+--+

|    role     |

+-------------+--+

| admin\_role  |

| test\_role   |

+-------------+--+

2 rows selected (0.251 seconds)

查看所有权限：

0: jdbc:hive2://master:10000/> show grant role admin\_role;

+-----------+--------+------------+---------+-----------------+-----------------+------------+---------------+-------------------+----------+--+

| database  | table  | partition  | column  | principal\_name  | principal\_type  | privilege  | grant\_option  |    grant\_time     | grantor  |

+-----------+--------+------------+---------+-----------------+-----------------+------------+---------------+-------------------+----------+--+

| \*         |        |            |         | admin\_role      | ROLE            | \*          | false         | 1461507543582000  | --       |

+-----------+--------+------------+---------+-----------------+-----------------+------------+---------------+-------------------+----------+--+

1 row selected (0.111 seconds)

0: jdbc:hive2://master:10000/> show grant role test\_role;

+------------+---------+------------+---------+-----------------+-----------------+------------+---------------+-------------------+----------+--+

|  database  |  table  | partition  | column  | principal\_name  | principal\_type  | privilege  | grant\_option  |    grant\_time     | grantor  |

+------------+---------+------------+---------+-----------------+-----------------+------------+---------------+-------------------+----------+--+

| sensitive  | events  |            | ip      | test\_role       | ROLE            | select     | false         | 1461558337008000  | --       |

| filtered   |         |            |         | test\_role       | ROLE            | \*          | false         | 1461557354579000  | --       |

+------------+---------+------------+---------+-----------------+-----------------+------------+---------------+-------------------+----------+--+

2 rows selected (0.111 seconds)

hive用户可以查看所有数据库、访问所有表：

0: jdbc:hive2://master:10000/> show databases;

+----------------+--+

| database\_name  |

+----------------+--+

| default        |

| filtered       |

| sensitive      |

+----------------+--+

3 rows selected (0.142 seconds)

0: jdbc:hive2://master:10000/> use filtered;

No rows affected (0.144 seconds)

0: jdbc:hive2://master:10000/> select \* from filtered.events;

+-----------------+----------------+----------------+--+

| events.country  | events.client  | events.action  |

+-----------------+----------------+----------------+--+

| US              | android        | createNote     |

| FR              | windows        | updateNote     |

| US              | android        | updateNote     |

| FR              | ios            | createNote     |

| US              | windows        | updateTag      |

+-----------------+----------------+----------------+--+

5 rows selected (0.458 seconds)

0: jdbc:hive2://master:10000/> select \* from sensitive.events;

+---------------+-----------------+----------------+----------------+--+

|   events.ip   | events.country  | events.client  | events.action  |

+---------------+-----------------+----------------+----------------+--+

| 10.1.2.3      | US              | android        | createNote     |

| 10.200.88.99  | FR              | windows        | updateNote     |

| 10.1.2.3      | US              | android        | updateNote     |

| 10.200.88.77  | FR              | ios            | createNote     |

| 10.1.4.5      | US              | windows        | updateTag      |

+---------------+-----------------+----------------+----------------+--+

5 rows selected (0.731 seconds)

#### 测试test\_role角色

使用test用户访问beeline：

beeline -u "jdbc:hive2://master:10000/" -n test -p test -d org.apache.hive.jdbc.HiveDriver

test用户不是管理员，是不能查看所有角色的：

0: jdbc:hive2://master:10000/> show roles;

ERROR : Error processing Sentry command: Access denied to test.Please grant admin privilege to test.

Error: Error while processing statement: FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.SentryGrantRevokeTask. SentryAccessDeniedException: Access denied to test (state=08S01,code=1)

test用户可以列出所有数据库：

0: jdbc:hive2://master:10000/> show databases;

+----------------+--+

| database\_name  |

+----------------+--+

| default        |

| filtered       |

| sensitive      |

+----------------+--+

test用户可以filtered库：

0: jdbc:hive2://master:10000/> use filtered;

No rows affected (0.256 seconds)

0: jdbc:hive2://master:10000/> select \* from events;

+-----------------+----------------+----------------+--+

| events.country  | events.client  | events.action  |

+-----------------+----------------+----------------+--+

| US              | android        | createNote     |

| FR              | windows        | updateNote     |

| US              | android        | updateNote     |

| FR              | ios            | createNote     |

| US              | windows        | updateTag      |

+-----------------+----------------+----------------+--+

5 rows selected (0.199 seconds)

test用户只能查看sensitive库中events表中的ip字段，其他的都都无法查看：

0: jdbc:hive2://master:10000/> use sensitive;

No rows affected (0.112 seconds)

0: jdbc:hive2://master:10000/> select \* from events;

Error: Error while compiling statement: FAILED: SemanticException No valid privileges

 User test does not have privileges for QUERY (state=42000,code=40000)

0: jdbc:hive2://master:10000/> select ip from events;

+---------------+--+

|      ip       |

+---------------+--+

| 10.1.2.3      |

| 10.200.88.99  |

| 10.1.2.3      |

| 10.200.88.77  |

| 10.1.4.5      |

+---------------+--+

5 rows selected (0.176 seconds)