Authorization in the Cloud: Enforcing Access Control Across Compute Engines

Hao Hao - hao.hao@cloudera.com
Li Li - lili@cloudera.com
About us

• Software Engineers @ Cloudera
• Working on Data Access Control projects
• Apache Sentry PMC and committer
Presentation Agenda

• Challenges for Authorization in the cloud
• Solution: Apache Sentry + RecordService
• Use Case + Demo
• Project Status
Challenges in the cloud
Moving to Cloud

• As cloud provides rapid access to flexible and low expense IT resources. Hadoop in Cloud becomes an increasingly common use case.

• “I can’t approve to buy hardware to expand my existing hadoop cluster, because we’ve got a CIO mandate to move IT to the cloud.”
Existing Cloud Provider Security

Simple “All or Nothing” permissions for each file
Challenges

“I have tables in S3 and user permissions assigned to files with IAM policy. However, the tables contain records from different, separately licensed sources. Only certain user groups are allowed to see certain records. “

How can I enforce this?

“In addition, we currently have multiple computing applications running on top of the same data, such as Spark, Hive, etc. “ How can we enforce the same access control policy?
Challenges

How to enforce authorization for structured data?

Access control for unstructured data, plus allow compute services above to access all structured data.
Challenges

“In our cloud scenarios, we have multiple storage engines involved - HDFS and S3. “

How can we be isolated from needing to know where our data is actually stored?
Demand for Fine-grained Authorization

Table level authorization

<table>
<thead>
<tr>
<th>Date/time</th>
<th>Account #</th>
<th>SSN</th>
<th>Asset</th>
<th>Trade</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:33:01 16-Feb-2015</td>
<td>3947848494</td>
<td>329-44-9847</td>
<td>TBT</td>
<td>Buy</td>
<td>EU</td>
</tr>
<tr>
<td>09:33:11 16-Feb-2015</td>
<td>0234837823</td>
<td>238-23-9876</td>
<td>AZP</td>
<td>Sell</td>
<td>US</td>
</tr>
<tr>
<td>14:12:34 16-Feb-2015</td>
<td>4848367383</td>
<td>123-56-2345</td>
<td>IDI</td>
<td>Sell</td>
<td>EU</td>
</tr>
</tbody>
</table>
Demand for Fine-grained Authorization

Column or row level authorization

<table>
<thead>
<tr>
<th>Date/time</th>
<th>Account #</th>
<th>SSN</th>
<th>Asset</th>
<th>Trade</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:33:01 16-Feb-2015</td>
<td>494</td>
<td>9847</td>
<td>TBT</td>
<td>Buy</td>
<td>EU</td>
</tr>
<tr>
<td>14:12:34 16-Feb-2015</td>
<td>383</td>
<td>2345</td>
<td>IDI</td>
<td>Sell</td>
<td>EU</td>
</tr>
</tbody>
</table>
Demand for Unified Authorization Enforcement

<table>
<thead>
<tr>
<th>Date/time</th>
<th>Account #</th>
<th>SSN</th>
<th>Asset</th>
<th>Trade</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:33:01 16-Feb-2015</td>
<td>494</td>
<td>9847</td>
<td>TBT</td>
<td>Buy</td>
<td>EU</td>
</tr>
<tr>
<td>14:12:34 16-Feb-2015</td>
<td>383</td>
<td>2345</td>
<td>IDI</td>
<td>Sell</td>
<td>EU</td>
</tr>
</tbody>
</table>
Demand for Storage and Compute layer Isolation
Solution: Apache Sentry + RecordService
Solution

- Business Intelligence
- Machine Learning
- ETC...

**COMPUTE**
- HIVE & IMPALA
- MR & SPARK

**STOREAGE**
- Kerberos & HDFS permissions
- AWS Access keys, IAM policies

- HDFS
- S3

- Record Service + Sentry
Apache Sentry

Authorization Service

● provides the ability to enforce role-based access control (RBAC) to data and/or metadata for authenticated users in a fine-grained manner.
● Enterprise grade big data security.
● Provides unified policy management.
● Pluggable and highly modular.
Apache Sentry

Work out of the box with Apache Hive, Hive metastore/HCataglog, Apache Solr, Apache Kafka, Apache Sqoop and Apache Impala
Apache Sentry

• Actors
  • User
  • User group membership
  • Role
  • Resources
  • Privilege
Apache Sentry

- User
  - Authenticated user
  - User identity obtained from session context

- User group membership
  - Defined outside of sentry policy
  - Obtained from user directory (LDAP, AD)
Apache Sentry

- Resources: is hierarchical
  - Data to be protected
  - Collection in Solr
  - Topic in Kafka
  - Columns in Hive
  - URI
Apache Sentry

- Privilege
  - Action or operation associated with a resource
    - SELECT on a given Column or Table
    - CREATE a TABLE or VIEW
    - QUERY on a SEARCH COLLECTION
    - Example: db=db1->table=t1->col=c1->action=SELECT
  - Assigned to a role
Apache Sentry

- Centralized Authorization Policies Store

```
grant select on col1 to role1
grant query on collection1 to role1
show role1
grant col1 --> role1
grant collection1 --> role1
```
Apache Sentry

Sentry provides fine-grained authorization on S3 as well as HDFS.
Apache Sentry + Record Service
Apache Sentry + Record Service

- Compute Engines
  - metastore
  - RecordService
  - RecordServicePlanner
  - Worker
  - Worker
  - Worker
  - Worker
  - Worker

- Cloud / On-premises Storage
Use Case
Fine-grained Authorization: Column + Row + Masking

• Given a table
  • CREATE TABLE s3_credit (credit_card_num STRING, name STRING, country STRING) STORED AS TEXTFILE LOCATION 's3a://recordservice-test-data/credit';
• Enforce same authorization policies on spark, impala and MR.
If Using Cloud Provider Security

Cloud providers only provide storage level permissions.

s3://user/credit  s3://user/credit_copy1  s3://user/credit_copy2  s3://user/credit_copy3
Using Sentry + RecordService

- Step 1:
  - Create Hive / Impala UDF: mask(String credit_card_num)
  - CREATE VIEW s3_credit_view AS SELECT mask(credit_card_num) masked_num, name name, country country FROM s3_credit where country='US';

```sql
select * from s3_credit_view
```

<table>
<thead>
<tr>
<th>credit_card_num</th>
<th>name</th>
<th>country</th>
</tr>
</thead>
<tbody>
<tr>
<td>8901</td>
<td>Alex</td>
<td>US</td>
</tr>
<tr>
<td>5155</td>
<td>Bob</td>
<td>US</td>
</tr>
<tr>
<td>1535</td>
<td>Mike</td>
<td>US</td>
</tr>
<tr>
<td>9890</td>
<td>Tod</td>
<td>US</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>masked_num</th>
<th>name</th>
<th>country</th>
</tr>
</thead>
<tbody>
<tr>
<td>********8901</td>
<td>Alex</td>
<td>US</td>
</tr>
<tr>
<td>********5155</td>
<td>Bob</td>
<td>US</td>
</tr>
<tr>
<td>********1535</td>
<td>Mike</td>
<td>US</td>
</tr>
<tr>
<td>********9890</td>
<td>Tod</td>
<td>US</td>
</tr>
</tbody>
</table>
Using Sentry + RecordService

- Step 2: Grant the Sentry privileges
  - CREATE ROLE test_role;
  - GRANT SELECT (name, country) on TABLE s3_credit to test_role;
  - GRANT SELECT on TABLE s3_credit_view to test_role;
  - GRANT ROLE test_role to GROUP testgroup;
select credit_card_num from s3_credit;

select * from s3_credit_view;
Spark

spark-shell --jars recordservice-spark-0.4.0-cdh5.8.x.jar

s3_credit

```scala
scala> val df = context.load("s3_credit", "com.cloudera.recordservice.spark")
warning: there were 1 deprecation warning(s); re-run with -deprecation for details
com.cloudera.recordservice.core.RecordServiceException: TRecordServiceException(code:INVALID_REQUEST, message:Could not plan request.,
detail:AuthorizationException: User 'testuser@HALXG.CLOUDERA.COM' does not have privileges to execute 'SELECT' on: default.s3_credit)
```

s3_credit_view

```scala
scala> val df = context.load("s3_credit_view", "com.cloudera.recordservice.spark")
warning: there were 1 deprecation warning(s); re-run with -deprecation for details

scala> df.collect.foreach(println)
[********8901,Alex,US]
[********5155,Bob,US]
[********1535,Mike,US]
[********9890,Tod,US]
```
MapReduce

```bash
hadoop jar recordservice-examples-0.4.0-cdh5.8.x.jar \
com.cloudera.recordservice.examples.mapreduce.RecordCount \
"select credit_card_num from s3_credit" "/user/testuser/tmp"
```

```
```

```bash
hadoop jar recordservice-examples-0.4.0-cdh5.8.x.jar \
com.cloudera.recordservice.examples.mapreduce.RecordCount \
"select * from s3_credit_view" "/user/testuser/tmp"
```

```
[li@vd0224 ~]$ hadoop fs -cat /user/testuser/tmp/part-r-000000
```

© Cloudera, Inc. All rights reserved.
Demo
Project Status
Project Status

Apache Sentry
- Graduated from Incubation – a top-level Apache project
- Hundreds of Cloudera customers using it

RecordService
- Open source project, and released up to Beta 0.3.0.
- Apache 2.0 Licensed
- Intent to donate to Apache Software Foundation
How to contribute?

● Mailing list:
  ○ recordservice-user@googlegroups.com
  ○ dev-subscribe@sentry.apache.org

● Contributions:
  ○ http://github.com/cloudera/RecordServiceClient/
  ○ https://cwiki.apache.org/confluence/display/SENTRY/Home

● Documentation:
  ○ http://recordservice.io/
  ○ https://cwiki.apache.org/confluence/display/SENTRY/Documentation
Q & A

Meet us @ Booth #721