Apache Sentry (incubating) and Hadoop Authorization

Sravya Tirukkovalur – sravya@apache.org
(Committer and PPMC)

Sept 29th 2015, Apache– Big data, Budapest
Hadoop Security

Different from traditional systems

1. All data in one place: HDFS/S3. No silos
3. There might be sensitive information in the ingested data.
Aspects of security

- **Authentication**
  - Is user “Bob” really “Bob”? 

- **Authorization**
  - Is user “Bob” allowed to do this operation?
  - Is user “Bob” allowed to see this data?

- **Encryption**
  - Restrict access by default.

- **Audit and governance**
  - Intrusion detection
  - Compliance requirements

- **Data lineage**
Existing authorization solutions

- Each compute framework has a dedicated authorization story.
  - Hive
  - HBase
  - HDFS
  - Sqoop
  - Spark

- They have overlapping underlying data
  - Hive databases, tables are accessible by HDFS/MR/Spark users.

- Duplicate policies => error prone and hard to get started and even harder to maintain
Sentry

• A secure service
• Provides unified policy management
• RBAC (Role based)
• Highly available
• Pluggable privilege model
Sentry Service

- Kerberos protected thrift server
- Gets user:group mapping from HadoopGroupMapping
- Supports many backend DBs
Plugins

- HS2
- HMS
- Impala
- Solr
- Sqoop
- HDFS
Sentry plugins

- Hive
- HiveServer 2
- Impala
- Catalog Service
- Hcat Plugin
- Sentry service
- Sentry
- JDBC app
- Pig users
- MR users
- REST App
HDFS sync

Single source of truth:
• User “Bob” has INSERT privileges on db1.tb1
• ACLs for “Bob” => WRITE_EXECUTE on /user/warehouse/db1.db/tb1
• Multiple compute frameworks can rely on same policies
Sentry plugin in Namenode

- Pig + HCAT
- MR + HCAT
- Sentry
- Sentry
- Sentry
- HCAT
- HS2
- Impala
- HMS (Hive meta store)
- HDFS
- /user/hive/warehouse

Other
HDFS plugin architecture

Translate to ACLS

Name node

Sentry

Name node

Sentry

Name node

Sentry

HMS

Sentry

Sentry Service

Path updates

Update logs for
• Path updates
• Perm updates
RBAC

Example:
- Role = Read_customer_data
- Groups (from LDAP)
  - Financial services – senior analysts
  - Product – senior analysts

- An organization-wide abstract. Regardless of how many domains, servers, and forests you might have, the RBAC system rides on top of all of it.
- Job-related. Roles represent job tasks
- A separation of duties. Whoever controls the underlying permissions shouldn’t control role membership, and vice-versa.

Reference:
- Why groups aren’t sufficient
- Economic analysis of RBAC
Generic model

• DB model:
  o Database
  o Table
  o View
  o Column

• Generic:
  o Resource type 1, resource value 1
  o Resource type 2, resource value 2 and so on
Import / Export of policies

- Import:
  sentry -command config-tool -I <filepath> -o

- Export:
  sentry -command config-tool -E <filepath>
Other features

• Column level access control:
  o Views were being used to enforce column level privileges.
  o `Read_restricted_customer_data` role = SELECT (id, zip_code) on table customer.

• Sentry HA
  o Uses curator framework
Record service integration

- Column level security for other compute engines
Apache community

Inclusive community which strives to achieve excellence through collaboration

- 5 releases
- 25 PPMC from diverse organizations
- 32 committers
- 21 contributors
- Diverse community: Female lead engineers
Roadmap

Some of them among many:

- Command line
- HDFS sync + HMS HA + Sentry HA
- Hive Auth V2
- HBase
- Kafka
- Gearing for graduation
Contributions welcome! 😊

- [https://cwiki.apache.org/confluence/display/SENTRY/How+to+Contribute](https://cwiki.apache.org/confluence/display/SENTRY/How+to+Contribute)
- Wiki: [https://cwiki.apache.org/confluence/display/SENTRY/Home](https://cwiki.apache.org/confluence/display/SENTRY/Home)
- Mailing list: [dev-subscribe@sentry.incubator.apache.org](mailto:dev-subscribe@sentry.incubator.apache.org)
Thank you!!!