# With Sentry, Cloudera Fills Hadoop’s Enterprise Security Gap

Every day, more data, users, and applications are accessing ever-larger Apache Hadoop clusters. Although this is good news for data driven organizations overall, for security administrators and compliance officers, there are still lingering questions about how to enable end-users under existing Hadoop infrastructure without compromising security or compliance requirements.

While Hadoop has strong security at the filesystem level, it lacks the granular support needed to adequately secure access to data by users and BI applications. Today, this problem forces organizations in industries for which security is paramount (such as financial services, healthcare, and government) to make a choice: either leave data unprotected or lock out users entirely. Most of the time, the preferred choice is the latter, severely inhibiting access to data in Hadoop.

Today, Cloudera is excited to launch Sentry, a new open source project that addresses these concerns. Sentry is an authorization module for Hadoop that provides the granular, role-based authorization required to provide precise levels of access to the right users and applications. Its new support for role-based authorization, fine-grained authorization, and multi-tenant administration allows Hadoop operators to:

* Store more sensitive data in Hadoop,
* Give more end-users access to that data in Hadoop,
* Create new use cases for Hadoop,
* Enable multi-user applications, and
* Comply with regulations (e.g., SOX, PCI, HIPAA, EAL3)

Sentry is now shipping as an add-on to CDH4.3 and will ship as a core component of CDH4.4 and Impala 1.1 and onward. Furthermore, we intend to nominate Sentry for the Apache Incubator to maximize its usefulness across the Hadoop ecosystem.

In the remainder of this post, we’ll offer more detail about why Sentry is needed and provide a technical overview of its capabilities and architecture.

**Hadoop Security, Before and After**

For Hadoop operators in finance, government, healthcare, and other highly-regulated industries to enable access to sensitive data under proper compliance, each of the four functional requirements must be achieved:

1. **Perimeter Security:** Guarding access to the cluster through network security, firewalls, and, ultimately, authentication to confirm user identities
2. **Data Security:** Protecting the data in the cluster from unauthorized visibility through masking and encryption, both at rest and in transit
3. **Access Security:** Defining what authenticated users and applications can do with the data in the cluster through filesystem ACLs and fine-grained authorization
4. **Visibility:** Reporting on the origins of data and on data usage through centralized auditing and lineage capabilities

Thanks to recent work in the Hadoop community (such as Cloudera’s [contribution of HiveServer2 to Hive](http://blog.cloudera.com/blog/2013/07/how-hiveserver2-brings-security-and-concurrency-to-apache-hive/)) as well as integration with solution providers, Requirements 1 and 2 are now addressed through Kerberos authentication, encryption, and masking. [Cloudera Navigator](http://blog.cloudera.com/content/cloudera/en/products/cloudera-enterprise/navigator-subscription.html) supports Requirement 4 via centralized auditing for files, records, and metadata. But Requirement 3, for access security, had been largely unaddressed, until Sentry.

***Access and Authorization without Sentry***  
Without Sentry, there are two suboptimal choices for authorization — coarse-grained HDFS authorization and advisory authorization — that do not meet typical compliance and data security needs for these reasons:

* Coarse-grained HDFS authorization: The primary mechanism of secure access and authorization is limited by the granularity of the HDFS file model. File-level authorization is coarse grained in that there is no ability to control access to the data within the file: a user either has access to everything in a file, or nothing. Furthermore, the HDFS permission model does not enable multiple groups to have different levels of access on the same data set.
* Advisory authorization: Advisory authorization is a rarely-used mechanism in Hive, designed to let benevolent users self-regulate against accidently deleting or overwriting production data. The system is “self-service” in that users can grant themselves any permission they’d like, as well as circumvent it. Thus, it doesn’t stop a malicious user from gaining access to sensitive data once they’ve been authenticated.

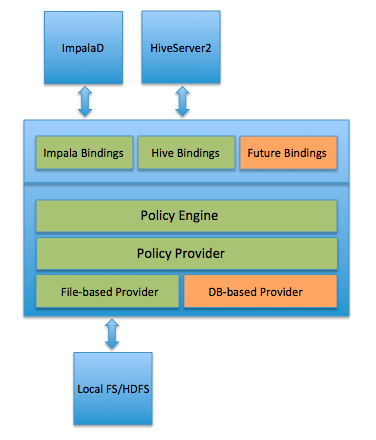
***Access and Authorization with Sentry***  
With the introduction of Sentry, Hadoop can now meet key RBAC (role-based access control) requirements for enterprise and government customers in these areas:

* **Secure authorization:** Sentry provides the ability to control and enforce access to data and/or privileges on data for authenticated users.
* **Fine-grained access control:** Sentry provides support for fine-grained access control to data and metadata in Hadoop. In its initial release for Hive and Impala, Sentry allows access control at the server, database, table, and view scopes at different privilege levels including select, insert, and all — allowing administrators to use views to restrict access to columns or rows. Administrators can also mask data within a file as required by leveraging Sentry and views with case statements or UDFs.
* **Role-based administration:** Sentry supports ease of administration through role-based authorization; you can easily grant multiple groups access to the same data at different privilege levels. For example, for a particular data set you may give your fraud detection team rights to view all columns, your analysts rights to view only non-sensitive or non-PII (personally identifiable information) columns, and your ingest processing pipeline rights to insert new data into HDFS.
* **Multi-tenant administration:** Sentry allows permissions on different data sets to be delegated to different administrators. In the case of Hive/Impala, Sentry allows administration of privileges at the level of a database/schema.
* **Unified platform:** Sentry provides a uniform platform for securing data; it uses existing Hadoop Kerberos security for authentication. Also, the same Sentry policy can be enforced while accessing data through either Hive or Impala. In the future, Sentry policy can also be extended to other components (more about that in the next section).

Next, we’ll explain how the Sentry architecture delivers these capabilities.

**Sentry Architecture**

Sentry is a highly modular and extensible mechanism. Initially, it allows Impala and Hive to enforce fine-grained security policies, but that capability can be extended to other frameworks, as well.



**Sentry architecture: Initial bindings are for Hive and Impala, with built-in extensibility to other frameworks.**

Sentry comprises a core authorization provider and a binding layer. The core authorization provider contains a policy engine, which evaluates and validates security policies, and a policy provider, which is responsible for parsing the policy. The binding layer provides a pluggable interface that can be leveraged by a binding implementation to talk to the policy engine. (Note that the policy provider and the binding layer both provide pluggable interfaces.)

At this time, we have implemented a file-based provider that can understand a specific policy file format. The policy file can reside either in the local filesystem or HDFS to get the benefits of replication and auditing. Although Cloudera has initially implemented support for Hive and Impala, it’s important to remember that the Sentry architecture is extensible: Any developer could implement a binding for a different component (such as Pig or Cloudera Search) or build a database provider that understands policies stored in a database-backed store.

The component-specific binding in Sentry implements a privilege model for the specific component and understands internal data structures. For example, the Hive binding implements a Hive-specific privilege model that allows fine-grained access to row/columns in a table as well as metadata operations such as show tables. (Impala’s model is very similar to that of Hive.)

**Conclusion**

We believe that Sentry is a major step forward in Hadoop security, making Big Data increasingly accessible by even more industries, organizations, and end-users – and giving administrators the flexibility, multi-tenant administration, and unified platform they need to make that happen easily. Cloudera is especially proud of the fact that we can not only contribute these new capabilities to the Hadoop ecosystem, but ship and support them inside our Big Data platform, Cloudera Enterprise.

Sentry is now available for [download](https://ccp.cloudera.com/display/SUPPORT/Downloads) as an add-on to CDH4.3, and you can explore the source code [here](https://github.com/cloudera/access) pending its Apache Incubator proposal status. Hive support is available through a base [Cloudera Enterprise](http://blog.cloudera.com/content/cloudera/en/products/cloudera-enterprise.html) subscription and Impala support through an [RTQ](http://blog.cloudera.com/content/cloudera/en/products/cloudera-enterprise/RTQ-subscription.html) subscription.