INTRODUCTION TO HADOOP SECURITY

Securing the Hadoop environment

When Hadoop was first released in 2007 it was intended to manage large amounts of web data in a trusted environment, so security was not a significant concern or focus. As adoption rose and Hadoop evolved into an enterprise technology, it developed a reputation as an unsecure platform. Most of the original Hadoop security shortcomings have been addressed in subsequent releases, but perceptions are slow to change. Hadoop's security reputation has outlasted its reality.

Security is actually quite inconsistent among Hadoop implementations because the built-in security and available options are inconsistent among release versions. It is also important to note that the commercial Hadoop distributions from software vendors (e.g. Cloudera, Hortonworks, MapR) have additional, proprietary security that is not included in the free Hadoop releases that are available from the Apache Foundation.

New Hadoop deployments can be extremely secure but many legacy Hadoop implementations may still have security gaps. This section highlights historical Hadoop security vulnerabilities and identifies the resources and tactics available to address them.

It is easy to quickly get lost in the details when talking about information security. To minimize confusion we will focus on three fundamental areas:

- How data is encrypted or otherwise protected while it is in storage (at rest) and when it is moving across the network (in motion)
- How systems and users are authenticated before they access data in the Hadoop infrastructure
- How access to different data is managed within the environment

(This article is part of our Hadoop Guide. Use the right-hand menu to navigate.)

Knox, Ranger simplify security management

The Hadoop ecosystem has resources to support security. Knox and Ranger are two important Apache open source projects. Knox provides a framework for managing security and supports security implementations on Hadoop clusters.

The Ranger project is focused on developing tools and techniques to help users deploy and standardize security across Hadoop clusters. It provides a centralized framework that can be used to manage policies at the resource level, such as files, folders, databases, and even for specific lines and columns within databases. Ranger helps administrators implement access policies by group, data type, etc. Ranger has different authorization functionality for different Hadoop components such as YARN, HBase, Hive, etc.

Knox is a REST API gateway developed within the Apache community to support monitoring, authorization management, auditing, and policy enforcement on Hadoop clusters. It provides a single access point for all REST interactions with clusters. Through Knox, system administrators can manage authentication via LDAP and Active Directory, conduct HTTP header-based federated identity management, and audit hardware on the clusters. Knox also supports enhanced security because it can integrate with enterprise identity management solutions and is Kerberos compatible.

Hadoop encryption

The original Hadoop release didn't include encryption. Later versions included end-to-end encryption that protects data while it is at rest within the Hadoop cluster and in motion across the network. In current releases all data stored in or accessible through HFDS is encrypted. Hadoop supports encryption at the disk, file system, database, and application levels.

In core Hadoop technology the HFDS has directories called **encryption zones**. When data is written to Hadoop it is automatically encrypted (with a user-selected algorithm) and assigned to an encryption zone. Encryption is file specific, not zone specific. That means each file within the zone is encrypted with its own unique **data encryption key (DEK)**. Clients decrypt data from HFDS uses an **encrypted data encryption key (EDEK)**, then use the DEK to read and write data. Encryption zones and DEK encryption occurs between the file system and database levels of the architecture.

Encryption keys need to be managed, which is the job of the Hadoop **Key Management Server**, or **KMS**. The KMS generates encryption keys, manages access to stored keys and manages encryption and decryption on HDFS clients. KMS is a Java web application with both client and server components that communicate with each other using HTTP and REST API. Security in KMS includes HTTPS secure transport and support for HTTP SPNEGO Kerboros authentication.

Hadoop distribution vendors and other solution providers have developed additional security that is either specific to their products or applicable to the entire Hadoop environment. Many enhancements focus on protecting Hadoop data while it is in motion. For example, there is the Wire encryption protocol that can applied to Hadoop data transmitted through HTTP, RPC, Data Transfer Protocol (DTP) and JDBC. There is also available SSL protection for JDBC clients and MapReduce shuffles, SASL for network RPCs and more.

Hadoop authentication

Authentication in the Hadoop environment has undergone a rapid and extensive evolution. The original Hadoop release did not include any provisions to authenticate users because it was a limited project intended for use in a trusted environment. A subsequent release added limited permission management features for files within the HDFS, but Hadoop still fell short of providing enterprise-class authentication security. Fast-forward to today, where the user authentication and identity management solutions that enterprises use for their core IT infrastructure can be extended to the Hadoop environment.

Today Hadoop is configurable in either secure or non-secure mode. The main difference is that secure mode requires authentication – for every user and service. Kerberos is the basis for authentication in Hadoop secure mode. Data is encrypted as part of the authentication process.

Many organizations perform authentication in the Hadoop environment by using their Active Directory or LDAP solutions. This approach formerly wasn't compatible with the Hadoop environment and is a good representation of how Hadoop is maturing and evolving. The Knox API from the Apache Hadoop project is used to extend Active Directory or LDAP to Hadoop clusters. It is also used to extend federated identity management solutions into the environment.

Hadoop access and permissions

Authenticating a user or service request does not automatically give it unrestricted access to all the data in the Hadoop cluster. Access rights can be set for portions of the HDFS and even for specific files and data types. As noted, **Ranger** facilitates the establishment and implementation of permission rights. Additional resources are also available. The **HDFS Permissions Guide** is a component that lets administrators set permissions for directories and files contained in the HFDS. Permissions can be set at the group and individual levels. Permissions include who can access the file, update it, delete it, etc. **Service Level Authorization** is a separate function that is used to verify that clients that attempt to connect to a specific Hadoop service are authorized to access it. Like the HDFS Permissions Guide, Service Level Authorization supports individual and group rights. **Sentry** is a module that works with Hive, HDFS data tables, Impala and other components, and is intended to provide more granular permissions management for data and metadata in Hadoop clusters.

Ranger, HDFS Permissions Guide, Service Level Authorization, and Sentry are part of the Hadoop project. Once again, additional permission protections and related security and management

features are built into the various Hadoop commercial releases.

Organizations also commonly execute **data masking** in Hadoop using various commercial solutions. Data masking refers to the practice of hiding original data records (through encryption) so they are not accessible to unauthorized users. Data masking is done in big data environments because many applications require some information from a dataset but not a complete record. For example, an imaging clinic may need to know how many patients received mammograms at one facility over a six-month period, but would not need to know the patient's results or complete medical histories. Patient outcome data would be relevant to a clinician, who would require different permissions.

Hadoop Security Then and Now

Component	Original Hadoop Release	Now Included/Available
Encryption	Not included	DEK encryption automatically applied to data in HFDS and in motion; additional data protection features are specific to each commercial distribution; KMS manages encryption keys; Kerberos is commonly used; additional encryption methods are Hadoop compatible/available.
Authentication	None	Kerberos is the foundation for Hadoop secure mode; Active Directory and LDAP extended to Hadoop; identity management solutions extended to Hadoop.
Access & Permissions	HDFS file permissions	Permissions can be set by individual, group, and role and set for specific data types and files; data masking can be applied to limit data that is accessed.

Solving big data security issues with Hadoop

Big data is by definition big, but a one-size-fits-all approach to security is inappropriate. The capabilities within Hadoop allow organizations to optimize security to meet user, compliance, and company requirements for all their individual data assets within the Hadoop environment. Capabilities like role, user, and group permissions, data masking, and multiple encryption and authentication options make it practical to provide different levels of security within a single, large environment. Growing integration support between Hadoop and Active Directory, LDAP and identity management solutions allows organizations to extend their enterprise security solutions so the Hadoop infrastructure doesn't have to be a silo.

Security is one of the fastest-changing aspects of Hadoop. The capabilities are continually enhanced and surpassed. For the latest security updates, check with the Apache project or your Hadoop distributor.