This article is an extract from *Performance*, Volume 8, Issue 1, February 2016. The full journal is available at **ey.com/performance**

Making security the foundation of big data infrastructure

Big data is driving invaluable new insights for organizations in all sectors. The sheer volume of data available demands new methods of analysis, drawing in many more collaborators. It also demands new, large-capacity storage. That combination presents risks, however: giving more users access to more data within a relatively new storage platform potentially makes that data vulnerable to hacking or leakage. Organizations should therefore take precautions to protect themselves.

This article is an extract from *Performance*, Volume 8, Issue 1, February 2016. The full journal is available at **ey.com/performance**

Authors

Abhay Raman
Partner, Cyber Risk Leader, EY, Canada

James Ki
Senior Manager, Enterprise Architecture, EY, Canada

Shezan Chagani
Manager, Cyber Risk Services, EY, Canada

This article is an extract from *Performance*, Volume 8, Issue 1, February 2016. The full journal is available at **ey.com/performance**

Big data describes data sets that are too large for conventional relational databases to process and manage. In the past few years, technology has developed that enables the analysis of large and disparate data sets to generate new insights and drive value within different functions of an organization.

Big data analytics can be leveraged as a strategic business asset, and this means organizations are moving away from purely governing and protecting their data to unlocking the value of the data collected within different parts of their organization.

Considerations for secure big data storage

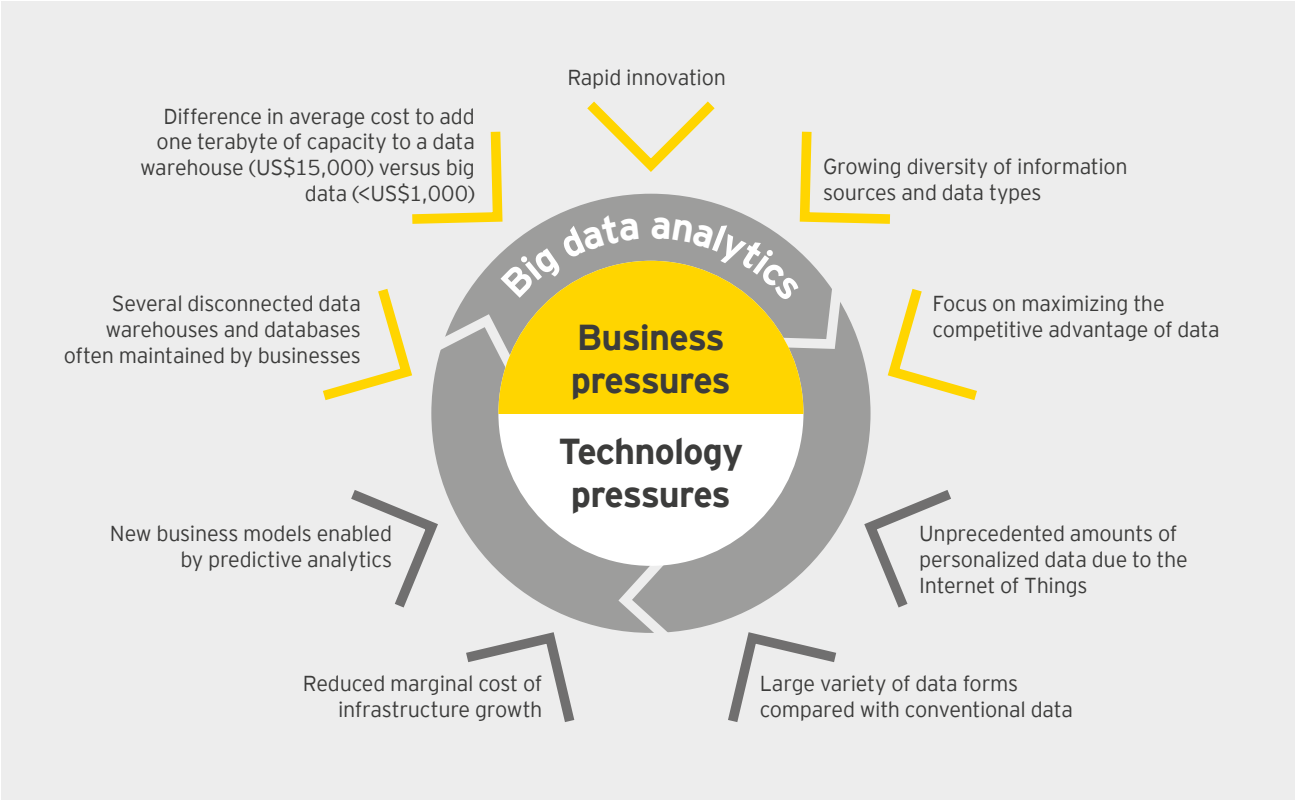
Data security needs to be a key focus – arguably more than ever. The fast-growing amount of data being generated, and the drive to reduce the cost of storing it, necessitates the adoption of big data infrastructure. Security needs within this infrastructure should immediately be prioritized as cyber attacks and data breaches impact more organizations worldwide.

A secured big data infrastructure should address some fundamental questions:

- Are people accessing data they shouldn't be? How do we know?
- How do we prevent leakage of sensitive data?
- How are we preventing accidental or deliberate unauthorized access?
- In a multi-tenant environment, how do we ensure groups only have the access they're authorized to have?

This article is an extract from *Performance*, Volume 8, Issue 1, February 2016. The full journal is available at **ey.com/performance**

Figure 1: Drivers of big data analytics



Securing a big data infrastructure based on Hadoop

Hadoop is one mainstream big data platform. It is an open-source software framework comprised of various components that interact with each other — and this structure can be vulnerable if not adequately secured.

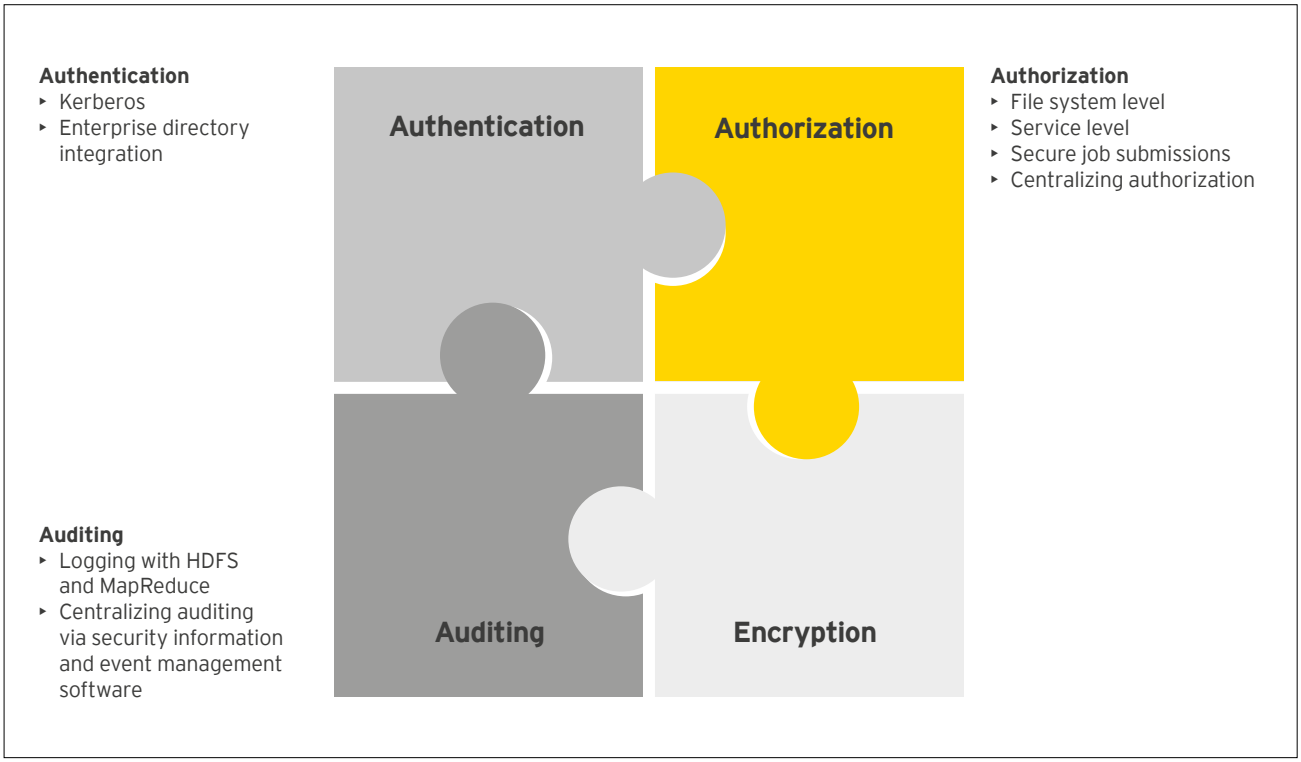
With our clients, we have noticed there is a tendency to focus primarily on perimeter security at the expense of neglecting internalized security. Sometimes, inconsistent authorization methods are implemented across the different Hadoop components, leading to oversights and missed configurations. It is also essential to consider critical security settings over and above Hadoop's default configurations.

There are four important areas to focus on when implementing Hadoop securely:

- Authentication
- Authorization
- Auditing
- Data encryption

This article is an extract from *Performance*, Volume 8, Issue 1, February 2016. The full journal is available at **ey.com/performance**

Figure 2: Framework to secure Hadoop infrastructure



Authentication

Initially, Hadoop was built within an open and trusted network, but its widespread adoption raised awareness that enhanced authentication was needed.

For example:

- A multistep process to access a file means multiple points of security must be controlled.
- Jobs are not run in real time, therefore security must be enforced to ensure components can run jobs on behalf of the requestor while maintaining traceability and accountability.
- Multi-tenancy within a single ecosystem introduces additional risks, as authentication needs to ensure users are properly segregated.

There are two recommended measures to strengthen authentication in the Hadoop ecosystem:

Implement Kerberos as the authentication mechanism
This provides greater security than using the Hadoop default simple authentication protocol. Kerberos is an authentication mechanism that allows communication through a ticketing system, and requires both services ("components") and users to be authenticated before gaining access to the cluster.

Integrate authentication with an Enterprise Directory
This enables centralized management and administration of user access, as well as single sign-on (SSO). Hadoop provides native integration with Enterprise Directory services, including Lightweight Directory Access Protocol (LDAP) and Active Directory (AD).

This article is an extract from *Performance*, Volume 8, Issue 1, February 2016. The full journal is available at **ey.com/performance**

Giving more users access to more data within a relatively new storage platform potentially makes that data vulnerable to hacking or leakage. Organizations should therefore take precautions to protect themselves.

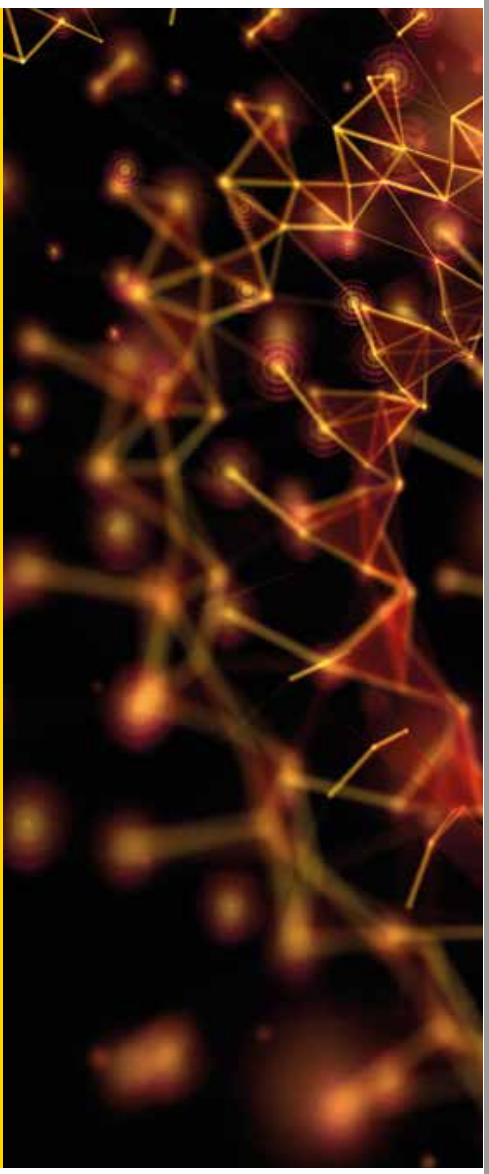
This article is an extract from *Performance*, Volume 8, Issue 1, February 2016. The full journal is available at **ey.com/performance**

Big data analytics can be leveraged as a strategic business asset, and this means organizations are moving away from purely governing and protecting their data to unlocking the value of the data collected within different parts of their organization.

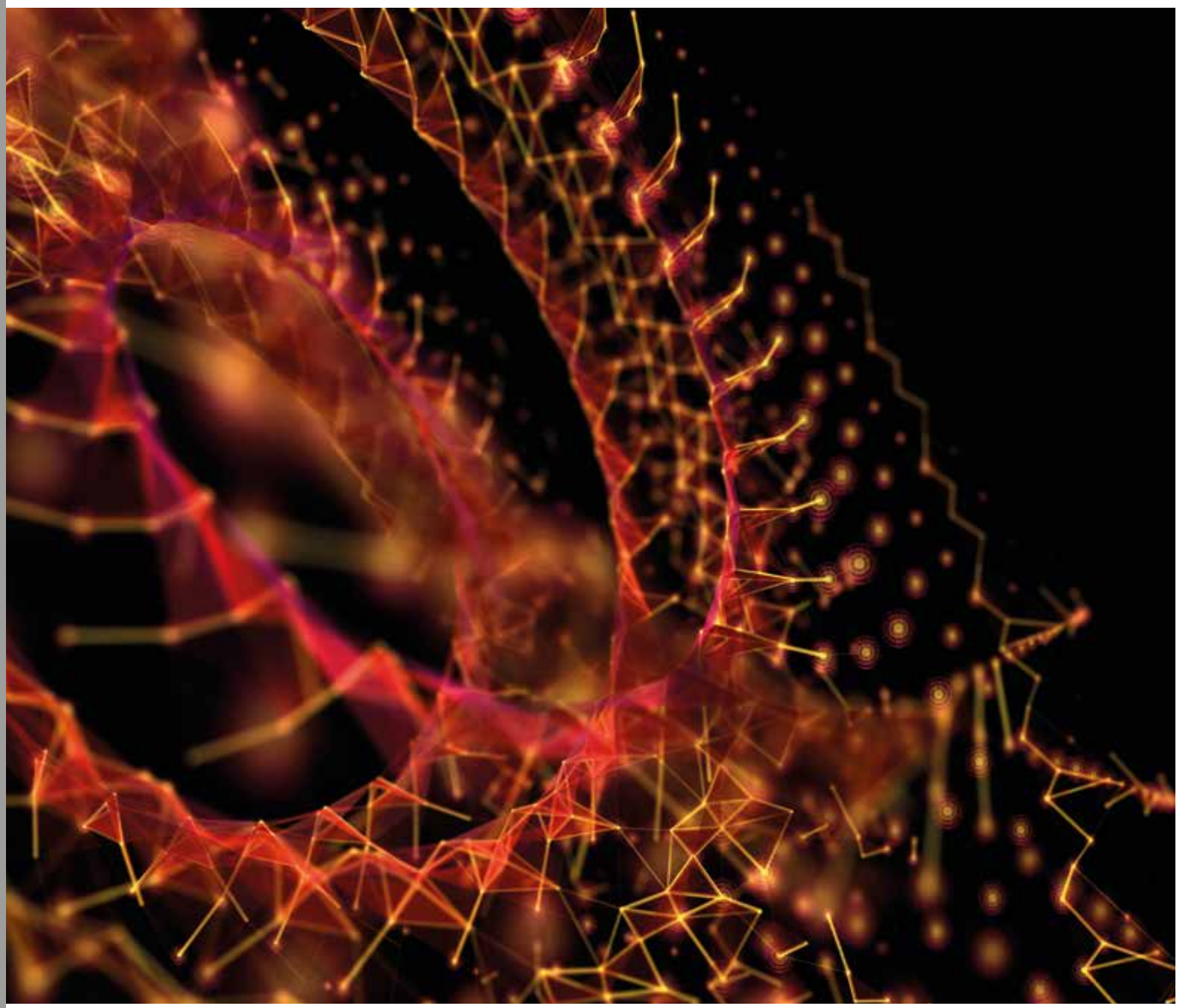
Client experience at a leading Canadian bank

The bank required assistance to design its security architecture for a fresh implementation of Hadoop 2.2, to support their credit risk projections. One of the many drivers was the ability to architect a system that supports multi-tenancy across groups that would be onboarded into the cluster. As part of their architecture, The bank aimed to have the maximum authorization configurations possible within Hadoop to support the demands of the business.

Key configurations included Linux Container Executor to separate job executions, strict group file system permissions, admin versus end-user group management permissions, strict service-level and user-level policies, and a fully secured, centralized access control list console to allow administrators to manage the cluster easily.



This article is an extract from *Performance*, Volume 8, Issue 1, February 2016. The full journal is available at **ey.com/performance**



This article is an extract from *Performance*, Volume 8, Issue 1, February 2016. The full journal is available at **ey.com/performance**

Initially, Hadoop was built within an open and trusted network, but its widespread adoption raised awareness that more enhanced authentication was needed.

Authorization

Authorization is one of the most critical parts of securing a Hadoop infrastructure and specifies the actions that authenticated users can perform.

Some key challenges include:

- Configuring the authorization to prevent access between departments
- Multilevel authorizations and queue management
- Managing the different configurations that are unique to each component, due to the different services that the component provides

Here are some recommended approaches for setting up and managing authorization within a Hadoop ecosystem:

- **Set permissions down to the file level.** Different groups within a shared Hadoop infrastructure can share the same file system. Enabling Hadoop Distributed File System (HDFS) permissions at file level ensures unauthorized users cannot access the data at rest and segregates data in a multi-tenant environment.
- **Configure service-level authorizations.** The service-level authorizations dictate the users and groups that are able to run jobs or services, so it is important to secure components that provide access to the data in addition to securing data at rest.
- **Limit access to job data solely to the user that requested it.** Secure jobs executed within the Hadoop ecosystem so that only the originator of the request has access to the output of their data. This is important as, by default, all Hadoop jobs are launched with the same system ID "yarn" – if compromised, this means anyone in the cluster can read all the data within it.
- **Implement a comprehensive queue modeling system.** While securing access to individual queues, set priorities to segregate access for different departments or user groups in a multi-tenant environment.
- **Centralize the management of authorization configurations.** Use an authorization manager, such as Apache Ranger, to reduce risk of inconsistent or out-of-date configurations across the Hadoop cluster.

Auditing

Auditing measures can actively help prevent security breaches – so they should be treated as more than a means of satisfying regulatory and security compliance. Auditing completes a security model by providing records of what has happened. For example:

- Active auditing can be used in conjunction with an alerting mechanism.
- Passive auditing refers to auditing that does not generate an alert.

The various components within a Hadoop ecosystem create disparate log files that become difficult to monitor and manage. Two steps to take in creating an audit baseline within a Hadoop ecosystem are:

- Ensure HDFS[1] and MapReduce[2] audit logs are adequately set to track both service and job activities at the file system level and at the compute layer.
- Ensure aggregate logs generated by different components within the Hadoop ecosystem are passed into a Security Information and Event Management (SIEM) application, such as IBM QRadar or Apache Ranger. This helps manage the vast volume of logs that are generated by the various Hadoop components and allow for correlation of events across components.

1. Hadoop Distributed FileSystem (HDFS): Java-based file system that provides scalable and reliable data storage, and was designed to span large clusters of commodity servers.
2. MapReduce: a programming model for processing and generating large data sets with a parallel, distributed algorithm on a cluster.

> The extent of the security configurations should always be tailored to each organization's needs and requirements, as the trade-off between performance and security will differ in each scenario.

Data encryption

No control is complete without applying data-level controls. There are two categories of data within Hadoop to focus on protecting:

- Sensitive data that has been loaded into Hadoop (business data or customer data) for analysis
- "Insights" – i.e., data that has already been analyzed. Such information, if exposed, can lead to great losses, as correlation has already been established

This data can flow within a Hadoop ecosystem as data at rest (in HDFS) or in motion (through the components). In either case, data can be secured by:

- Setting encryption zones across a file system and leaving other areas unencrypted, in order to achieve balance between performance and security in a multi-tenant environment. A Hadoop cluster can easily contain Petabytes of data; therefore, securing this data is critical due to the wealth of information that is stored.
- Securing access across end point connections, as the connections can provide an entry point for attacks to penetrate the Hadoop ecosystem. For example:
  - At the client level, secure the clients communicating to HDFS through remote procedure calls and data transfer protocol.
  - In user mechanisms, secure the browser level and command line interfaces via HTTPS and JDBC[3] measures.
  - To secure the shuffle, apply HTTPS during data exchange in the core components of analytics.

Conclusion

Big data analytics presents both big opportunities and challenges for businesses. Securing a big data infrastructure introduces further challenges that are not applicable to traditional relational database technologies and, as such, a structured approach should be taken.

It is important to note that the extent of the security configurations should always be tailored to each organization's needs and requirements, as the trade-off between performance and security will differ in each scenario. Although there are many more configurations that can be applied, this article highlights just some of the key areas that should be prioritized when securing a big data infrastructure. ■

3. Java DataBase Connectivity (JDBC): an application program interface (API) for the Java programming language that provides methods for querying and updating data in a database.