EXECUTIVE SUMMARY

Everyday, businesses are seeing multiple high-profile cyber attacks across the globe, generating tremendous pressure to create and maintain a strong line of defense in their protection of data. The Hacker News has reported severe problems with Hadoop: "Nearly 4,500 servers with the Hadoop Distributed File System (HDFS) were found exposing more than 5,000 terabytes of data, according to an analysis conducted using Shodan search engine. This exposure is due to the same issue – HDFS-based servers, mostly Hadoop installs, haven’t been properly configured."

In many of these high profile attacks, the common thread is the fact that business and consumer data was compromised – consumers’ credit card number, social security number, banking information, etc. – which highlights the importance of data protection.

The MapR out-of-the-box protection scheme is designed to maintain a strong line of defense in the protection of data, which is the focus of this paper.
OVERVIEW

The MapR approach is a data platform with a data protection scheme built directly into the platform and enabled by default. The secure MapR Data Platform is the only data platform designed with security out-of-the-box. The platform offers a robust and unmatched protection scheme for the data within it. The MapR security model is built directly into the platform, supporting the ability to apply security protection directly as data comes into and out of the platform without requiring an external security management server or a particular security plug-in into each ecosystem component. MapR security semantics are applied automatically by design for data being retrieved or stored by any ecosystem, application, or users, out-of-the-box.

Our complete data protection includes:

Secure by Default. A unique way to ensure your data platform is properly secure, MapR now is secure out-of-the-box, which includes the MapR Data Platform and Ecosystem for new installations with a single click to disable if desired. All network connections are encrypted with authentication enabled, and all data is stored encrypted.

Platform-Based Security. The only data platform with built-in security designed to apply security semantics automatically as data is being stored and retrieved from the platform. Solve for all four pillars of security (authentication, authorization, auditing, and encryption), using platform-level capabilities that don't require external security tools or plugins. Such a solution is therefore complete and cannot be bypassed by components that have not been carefully altered to work with an external security tool.

Encryption. Data is protected by encrypting all data being transmitted over the wire and encrypting all that is stored in the MapR Data Platform.

Data Governance. The MapR DataOps Governance Framework is built on an open architecture, allowing customers to extend and use the right technology to support their processes that match their use cases, which can track and manage the full data transformation process to achieve a complete data governance and data lineage monitoring solution.

Data Lifecycle Management. Ability for customer to have full control over the ability to place data on different nodes with different performance characteristics as well as offload data for archival purposes.

SECURITY BY DESIGN

Security by design is the MapR approach, seeking to make every effort to be free of vulnerabilities and impervious to attacks with built-in safeguards and adherence to best programming practices. While it is difficult to design software to be 100% vulnerability-free, MapR is focused on minimizing the more common mistakes in software engineering by embracing a wide range of industry security principles, focused on building security directly into the product. Some of the critical tenets built into our platform are:

Principle of Defense in Depth. One of the MapR hallmark security capabilities is focused on having security at each layer of the architecture, making it more difficult and unlikely for an intruder to compromise your data. The core of MapR Security is built directly into the platform, supporting the ability to apply security protection directly as data comes into and out of the platform without requiring an external security manager server or a particular security plug-in into each ecosystem component. In conjunction with the core platform, the MapR Ecosystem will offer an additional layer of security, leveraging robust MapR Native Security, featuring ease-of-use security capabilities.
Establish Secure Defaults. MapR has embraced the secure by default principle with the behavior goal to be secure out-of-the-box with an option allowing the user to reduce the security if appropriate. It provides comfort, knowing the platform is secure without relying on complex, brittle scripts and complicated documentation steps, hoping nothing went wrong along the way.

Keep Security Simple. MapR ease-of-use security capabilities bring a simple yet robust approach to security, which prevents users from making mistakes configuring security. This is accomplished by an innovative, alternative Kerberos design called MapR Native Security. Kerberos is well-known in the industry as being very robust but also very difficult for many users to understand, manage, and correctly configure. There is nothing simple about Kerberos. For this reason, MapR Native Security was invented and implemented in leveraging the robustness of Kerberos but simplifying the complexities. In particular, MapR self-manages the keys used to encrypt and sign tickets, instead of relying on Kerberos keytab files, which are very difficult for most users to maintain.

Separation of Duties. The MapR Data Platform has been designed to support the ability for MapR administrators to administer the platform while not being allowed to see any user data.

Fail-Safe Security. The MapR Data Platform is designed not to allow unintended access due to an exception or failure in the system.

Detect Intrusion. MapR auditing generates an audit record for all data access as well as audit records for any authorization denials. The audit data is sent to MapR-ES, allowing customers to send MapR security-relevant auditing data into the user’s intrusion detection solution that is available in the enterprise.

Standardized Integration. MapR use Linux account integration via PAM, which is a standard functionality to authenticate users and access their user/group information. This is the same function any user employs when logging into a Linux box. As such, MapR will work with any enterprise user registry that supports PAM with minimal configuration.

MAPR CORE SECURITY FEATURES

AUTHENTICATION

[Diagram showing authentication process]
**User-based authentication:** MapR leverages the Linux Pluggable Authentication Modules (PAM), which give the broadest range of registry support for authenticating a user to a MapR cluster.

**System-based authentication:** MapR supports both MapR Native Security and Kerberos.

**Kerberos.** A commonly used protocol for authenticating (i.e., identifying) users on a computer system, including Hadoop clusters. Kerberos is a ticket-based system in which the user first requests a ticket from the Kerberos server, and the issued ticket is used as a trusted identifier to all services covered by that Kerberos server. The Kerberos integration with MapR lets you leverage your existing Kerberos infrastructure for authenticating users on your MapR cluster.

**MapR Native Security.** An alternative to Kerberos, MapR provides a native authentication mechanism that operates equivalently to Kerberos but offers a much-simplified configuration, eliminating the need to manage the Kerberos keytab files, due to the MapR system managing the keys automatically. Similar to Kerberos, the user will first request a ticket from the MapR CLDB server, using a username and password that is validated, leveraging PAM interface, and then the ticket is used as a trusted identifier to all the services within MapR.

For more detail information on MapR Security, see [MapR Authentication Architecture](#).

**MapRlogin Utility.** The maprlogin utility supports user authentication with either username and password or Kerberos to generate a unique session token called a ticket. The following diagram outlines the process flow:
**Authentication Flow**

On clusters that use Kerberos for authentication, a MapR ticket is implicitly obtained for a user that runs a MapR command without first using the maprlogin utility. The implicit authentication flow for the maprlogin utility first checks for a valid ticket for the user, then uses that ticket if it exists. If a ticket does not exist, the maprlogin utility checks if Kerberos is enabled for the cluster, then checks for an existing valid Kerberos identity. When the maprlogin utility finds a valid Kerberos identity, it generates a ticket for that Kerberos identity.

When you explicitly generate a ticket, you have the option to authenticate with your username and password or authenticate with Kerberos:

1. The user on the client machine invokes the maprlogin utility, which connects to a CLDB node in the cluster using HTTPS. The hostname for the CLDB node is specified in the mapr-clusters.conf file.
   a. When using username/password authentication, the node authenticates using PAM modules with the Java Authentication and Authorization Service (JAAS). The JAAS configuration is specified in the mapr.login.conf file. The system can use any registry that has a PAM module available.
   b. When using Kerberos to authenticate, the CLDB node verifies the Kerberos principal with the keytab file.

2. After authenticating, the CLDB node uses the standard UNIX APIs getpwnam_r and getgrouplist, which are controlled by the /etc/nsswitch.conf file, to determine the user’s user ID and group ID.

3. The CLDB node generates a ticket and returns it to the client machine, completing the login communication between the client and the CLDB.

4. After login, the client communicates with a MapR server. The server validates that the ticket is properly encrypted, to verify that the ticket was issued by the cluster’s CLDB.

5. The server also verifies that the ticket has not expired or been blacklisted.

6. The server checks the ticket for the presence of a privileged identity, such as the mapr user. Privileged identities have impersonation functionality enabled.

7. The ticket’s user and group information are used for authorization to the cluster, unless impersonation is in effect.

**AUTHORIZATION**

![Diagram of the authentication process]

Can you do this?
MapR provides sophisticated authorization controls to ensure that users can perform only the activities for which they have permissions, such as data access, job submission, cluster administration, etc. These permissions can be granted by an administrator via the browser-based MapR Control System (MCS) management and monitoring interface or command line utilities.

Unix File Permissions. For files and directories in the MapR Data Platform, you can leverage standard Unix-style permissions to grant access to authorized users. Since the MapR Data Platform is a POSIX file system with full read/write capabilities, it can be accessed the same way that Linux file systems are accessed. This means existing file-based Linux applications can access files in MapR without any code changes or recompilation.

Access Control Expressions (ACEs) are a powerful and flexible mechanism to grant permissions on structured and unstructured data. With ACEs, you get more flexibility than standard access control lists (ACLs). ACEs are Boolean expressions that allow AND and OR logic when defining permissions. The flexibility lets you specify fine-grained access control at the column and/or column-family level in MapR-DB, an HBase binary and document database. Examples of ways you can grant permissions include:

- OR-based permissions found in standard ACLs
  - "Sales department" OR "marketing department"

- AND-based permissions
  - "VP level" AND "marketing department"

- Granular permissions
  - ("VP level" OR "director level") AND ("sales department" OR "marketing department") AND ("John Doe")

For Files and Directories, an ACE allows you to define access (whitelist and blacklist) to files and directories for a combination of users, groups, and roles. If ACEs are not set, POSIX mode bits for the file or directory will be used to grant or deny access to the file or directory. For more information, see File and Directory ACEs.

For Volumes, an ACE allows the whole volume to define whitelists (to grant access) and blacklists (to deny access) for files and tables within a volume. For more information, see Whole Volume ACEs.

For MapR-DB, an ACE is used exclusively to set permissions for MapR tables, column families, and columns. For more information, see Enabling Table Authorization with ACEs.

For MapR-ES, a Kafka API-Based pub/sub system, an ACE is used exclusively to set permissions for utilities. For more information, see MapR-ES Utilities.

Access Control Lists. MapR supports access control lists (ACLs) to grant permissions for performing administrative tasks at both the cluster and the volume level. Examples of tasks include starting/stopping services, creating volumes, creating mirrors, and changing mirror properties. MapR ACLs also control which users and groups can perform specified tasks on specified job queues, including the ability to submit, kill, or reprioritize jobs. For more information, see Managing Access Control Lists.
AUDITING

The auditing capabilities in MapR are critical for regulatory compliance as well as for understanding user behavior in the system. Regulations often require the ability to prove which user accessed which data, and logging user behavior helps in several situations, including identifying suspicious activities on sensitive data.

MapR records access of data (files, directories, and MapR-DB table data) that are enabled for auditing as well as operations on these objects and executions on the command line (maprcli), including those commands that modify the configurations of a MapR cluster. Log entries are streamed in real time to MapR-ES, written in JSON format, and can be analyzed with Apache Drill, your security information and event management (SIEM) solution, or other third party tools. Log files are also retained for as long as you specify.

MapR auditing consists of:

• All admin activities via maprcli, REST, or the MapR Control System (MCS)
• Authentications to MCS
• Operations on directories and files
• Operations on MapR-DB tables
• Operations on MapR-ES

In addition, each ecosystem offers operational auditing.

For more information on auditing, see Managing Auditing.
MapR supports data encryption as an additional means of preventing unauthorized access to sensitive data. Encryption is used to avoid exposure to breaches, such as packet sniffing and theft of storage devices.

**Over-the-Wire Encryption.** To avoid data theft by packet sniffing, over-the-wire encryption is available between MapR nodes and between a MapR cluster and ecosystem. For more information, see Encryption Architecture: Wire-Level Encryption.

**Encryption at Rest.** Encryption at rest not only prevents unauthorized users from accessing sensitive data, but it also protects against data theft via sector-level disk access. If data at rest encryption (DARE) is enabled, MapR automatically encrypts data at rest and manages the keys used to encrypt data seamlessly – no need for special utilities to encrypt or decrypt the data. New volumes are encrypted by default with the option to disable during volume create.

**Field-Level Encryption.** Enables securing specific sections of data residing in files. This capability logically behaves like access controls on a structured data set in a database management system. Some data elements in the files will remain open, while the secured data elements will be encrypted and can be decrypted by authorized users when used in conjunction with key management technologies. Field-level encryption is provided by MapR Advantage Partners specializing in data security, such as Dataguise or Informatica.

**Format-Preserving Encryption (FPE) and Masking.** This is a mechanism for encrypting data, so that the format remains the same. This allows applications to access data that looks legitimate, instead of the typically garbled text that encryption outputs. This technique is particularly useful for analytical tasks that require readability in the encrypted data elements. Masking is similar in that it replaces sensitive data elements with an unidentifiable value, but it is not truly an encryption technique, so the original value cannot be returned from the masked value.

A significant benefit of these techniques is that the cost of securing a big data deployment is reduced. As secure data is migrated from a secure source into your platform, FPE or masking reduces the need for applying additional security controls on that data while it resides in your platform.

Both of these techniques are available from MapR Advantage Partners specializing in data security, such as Dataguise or Informatica.
SECURITY PROTOCOLS USED BY MAPR

For information on specific security protocols supported by different components, see the Security Support Matrix.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Encryption</th>
<th>Authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>MapR RPC</td>
<td>AES/GCM</td>
<td>maprticket</td>
</tr>
<tr>
<td>Hadoop RPC and MapR-SASL</td>
<td>AES/GCM</td>
<td>maprticket</td>
</tr>
<tr>
<td>Hadoop RPC and Kerberos</td>
<td>Kerberos</td>
<td>Kerberos ticket</td>
</tr>
<tr>
<td>Generic HTTP Handler</td>
<td>HTTPS using SSL/TLS</td>
<td>maprticket, username and password, or Kerberos SPNEGO</td>
</tr>
</tbody>
</table>

HTTPS EXCLUDED CIPHERS

By default, the following weak TLS/SSL ciphers are excluded from MapR HTTPS implementation:

- SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
- SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
- SSL_RSA_EXPORT_WITH_RC4_40_MD5

You can modify this list of excluded ciphers by editing the hadoop.ssl.exclude.cipher.suites property in the core-site.xml file. Restart the web servers that use the HTTPS protocol after changing the list of excluded ciphers. The following web servers use HTTPS:

- MCS
- JobTracker
- taskTracker
- Node Manager
- ResourceManager
- HistoryServer
- CLDB
- HBase

CONCLUSION

Ensuring that business data is efficiently and securely managed begins with a data platform that is appropriately designed from the ground up. MapR delivers security out-of-the-box and offers ease-of-use security capabilities without compromising results in high data quality, integrity, and trustworthiness for a better business outcome. The MapR Data Platform is designed with a robust and unmatched data protection scheme built directly into the platform. No external or open source security manager server with security plugins into each ecosystem component is required. MapR security semantics are applied automatically by design for data being retrieved or stored by any ecosystem, application, or users out of the box.

No other data platform on the market can match the security capabilities offered.