Securing Drill

MapR 6.0 introduces a default security configuration that automatically secures the MapR cluster and ecosystem components when you install them manually or using the MapR Installer.

The default security configuration uses MapR security (mapr tickets) to provide authentication, authorization, and encryption for cluster security. By default, security is configured automatically for Drill when you install Drill (../AdvancedInstallation/drill_installation.html#installing_drill) 1.11 on secured MapR 6.0 clusters.

Impact of MapR Default Security on Drill

The default MapR installation with security configured has the following impact on Drill:

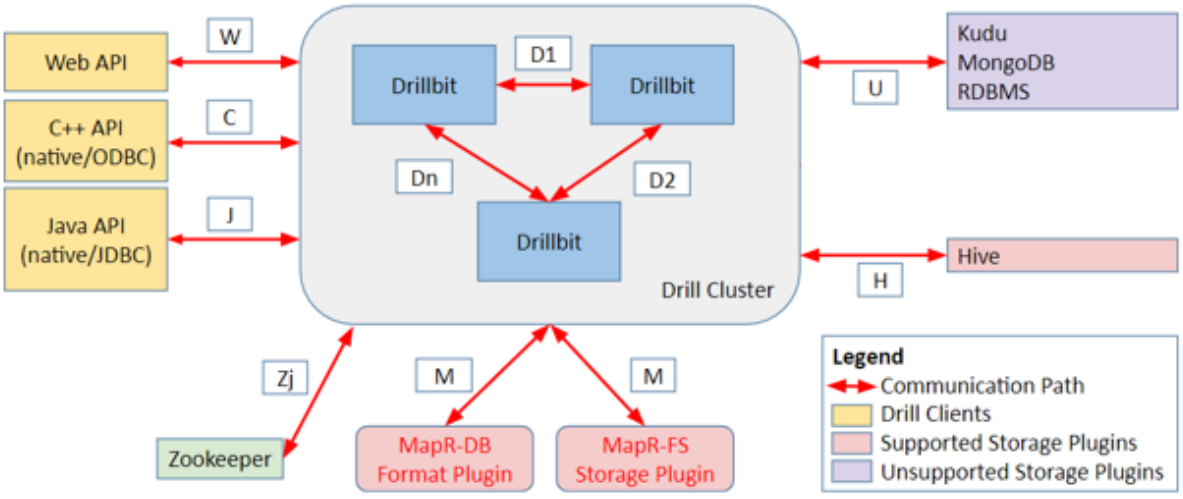
- Simplifies the process of securing the cluster. The default security configuration automatically secures all communication channels, as described in the *Drill Security Architecture* section below.
- Introduces new Drill configuration files; in addition to drill-override.conf, distrib-env.sh, and drill-env.sh, there is now a drill-distrib.conf file.

  Note: Modifying drill distribution-specific files is highly discouraged. To customize any Drill configuration, use drill-override.conf and drill-env.sh.

- May affect the performance of Drill. See the *Impact of Default Security on Drill Performance* section below.

Drill Security Architecture

Drill 1.10 and later supports authentication using Plain, Kerberos, and MapR-SASL to secure the network channel between the Drill client and Drillbit, and also between Drillbits. As of Drill 1.11, the default MapR security configuration secures all network channels with MapR Security (configure_server_auth.html#configure_server_auth) (except for HTTPS, which uses SSL (config-ssl-encrypt.html#config-ssl-encrypt)), as shown in the following image:



The following table lists the types of security supported for each communication path in the diagram above, as well as the components involved in the communication:

Note: MapR security supports authorization using the file system capabilities, such as ACEs. See Configuring MapR Security (../SecurityGuide/Configuring-MapR-Security.html#Configuring-MapR-Security).

Type of Security Supported	Communication Path	Component Communication

Authentication and encryption using MapR security	C	ODBC client/C++ API to Drillbits
	J	JDBC client/Java API to Drillbits
	D1, D2, Dn	Drillbit to Drillbit
	M	Drillbit to MapR-DB/MapR-FS
	H	Drillbit to Hive **Note:** You must manually configure Hive security. See Configuring the Hive Storage Plugin (configure_hive_storage_plugin.html#ConnectingApacheDrilltoDa_26986462-d3e302).
Plain authentication with SSL encryption (HTTPS enabled)	W	Web client/Web API to Web server **Note:** The HTTPS channel (Web client) uses Plain authentication to authenticate a Web client with SSL/TLS for encryption. This is configured by default in a secure 6.0 cluster with Drill 1.11 installed. See Using SSL/TLS for Encryption (config-ssl-encrypt.html#config-ssl-encrypt)
Authentication with MapR security, no encryption	Zj	Drillbit to ZooKeeper **Note:** The Drillbit creates znodes, for which ZooKeeper ACLs provide security. See the *Zookeeper ACLs for Drill Security* section below.

Note:

- Kerberos is not enabled or configured as part of the default security configuration.
- As of MapR 6.0 and Drill 1.11, HBase is no longer supported, therefore the communication path between Drill and HBase is also not supported.
- Drill clients running versions of Drill prior to 1.11 cannot connect to Drillbits installed using the default security configuration.
- The Drillbit Web server uses SSL certificates provided by the default MapR installation. See Using SSL/TLS (config-ssl-encrypt.html#config-ssl-encrypt) for Encryption.

Limitations

- If you unsecure a cluster, you must backup the Drill znodes. After the switch to unsecured, update the ACL on the Drill znodes so that Drill in an unsecured cluster can access all Drill znodes. See the *Security Between ZooKeeper and Drillbits* section below.
- The Hive storage plugin is not secured by default and requires that you manually modify the storage plugin configuration to enable security. See Configuring the Hive Storage Plugin (configure_hive_storage_plugin.html#ConnectingApacheDrilltoDa_26986462-d3e302).

Security Between ZooKeeper and Drillbits

When the MapR cluster is running MapR 6.0 or later and Drill 1.11 or later with the default security enabled, authentication is enabled between the Drillbits and ZooKeeper, by default, and the ZooKeeper znode information is secured automatically through authentication and znode ACLs (https://zookeeper.apache.org/doc/r3.1.2/zookeeperProgrammers.html#sc_ZooKeeperAccessControl).

Note: The communication between Zookeeper and the Drillbits is not encrypted.

Drill uses ZooKeeper to discover and coordinate Drillbits. Drillbits use Zookeeper znodes to store coordination-related information, as well as execution-related information. If the information stored in the znodes is not properly secured, it can compromise the security and/or privacy of the cluster.

The znode ACLs are set such that only the Drillbit user (the user that started the Drillbits in the cluster) can access (create, delete, read, write, administer) all of the Drillbit ZooKeeper nodes, with the exception of the Drill ZooKeeper nodes that hold information about the Drillbits in the cluster. Drill clients use these Drill ZooKeeper nodes to discover and connect to the Drillbits in a cluster.

Note: If you installed Drill 1.11 on a MapR cluster that does not have the default security configuration, and you are configuring custom security, you must enable authentication and set ACLs on the znodes manually (https://zookeeper.apache.org/doc/r3.1.2/zookeeperProgrammers.html#sc_ZooKeeperAccessControl). However, setting ACLs manually for Drill znodes is not recommended. If you set ACLs manually, you must set them the same as a Drillbit would have set them in a secure MapR cluster, as described previously.

For additional information and instruction, refer to the following pages:

- https://zookeeper.apache.org/doc/trunk/zookeeperOver.html (https://zookeeper.apache.org/doc/trunk/zookeeperOver.html)
- https://zookeeper.apache.org/doc/r3.1.2/zookeeperProgrammers.html#sc_ZooKeeperAccessControl (https://zookeeper.apache.org/doc/r3.1.2/zookeeperProgrammers.html#sc_ZooKeeperAccessControl)
- https://cwiki.apache.org/confluence/display/ZOOKEEPER/Zookeeper+and+SASL (https://cwiki.apache.org/confluence/display/ZOOKEEPER/Zookeeper+and+SASL)

Drill Configuration Files

The Drill installation includes configuration files with start-up options that you can modify prior to starting Drill. The configuration files reside in a HOCON (https://github.com/typesafehub/config/blob/master/HOCON.md) configuration file format, which is a hybrid between a properties file and a JSON file. The files have a nested relationship and a hierarchical structure, where one file overrides another. You can locate the files in the /opt/mapr/drill/drill-<version>/conf directory.

The configuration files are listed below in their hierarchical order. The drill-distrib.conf file overrides the drill-module.conf file, and the drill-override.conf file overrides the drill-distrib.conf file.

- drill-override.conf
- drill-distrib.conf
- drill-module.conf

Environment variables are also overridden in the same way, in the order listed below:

- drill-env.sh (or explicitly defined in environment)
- distrib-env.sh
- drill-config.sh

The following table lists the configuration files and their descriptions:

File Name	Description	Default Config with Secure Installation

drill-distrib.conf	Introduced in Drill 1.11, this file contains MapR distribution specific configurations for Drill. Automatically updated by configure.sh when you configure the MapR cluster.	Enables authentication, impersonation, and encryption with default mechanisms as Plain and MapRSasl. Enable TLS for Https channel. By default it will use SSL certificate provided by the MapR installation, which can be updated by admin to use a specific certificate in a key-store.
distrib-env.sh	Contains distribution-specific defaults for various environment variables.	Enables authentication between the Drillbits and ZooKeeper. Configures the location of the MapR security configuration file, mapr.login.conf, used by Drill.
drill-env.sh	The drill-env.sh file contains the cluster admininstrator-specific environment variables that can differ from the defaults. You can modify this file to override the default values of system properties defined in the distrib-env.sh file or to define a new system property.	Empty upon installation.
drill-override.conf	Use the drill-override.conf file to override the default values obtained from drill-module.conf and drill-distrib.conf. A cluster administrator can update this file to configure a Drillbit as required (different from default installation)	When you first install Drill, drill-override.conf contains ZooKeeper and Drillbit configuration information, however after you run configure.sh -R, the entries are removed and the file does not contain any configurations.

Impact of MapR Default Security on Drill Performance

The default MapR secure configuration enables encryption for all network channels, which can affect how Drill performs. If performance is your highest priority, you can install MapR and Drill without security enabled, and have your security expert manually configure cluster security.

Note: Running configure.sh with the `-unsecure` parameter, as shown, turns security off in the entire MapR cluster:

```
/opt/mapr/server/configure.sh -forceSecurityDefaults [ -unsecure | -secure ]
 -C <CLDB_node> -Z <ZK_node>
```

Alternatively, you can install MapR and Drill with security enabled, and disable individual Drill security settings. For example you can modify the drill-override.conf file and disable encryption, leaving authentication enabled.

See Drill Installation (../AdvancedInstallation/drill_installation.html#installing_drill) and configure.sh (../ReferenceGuide/configure.sh.html#configure.sh) for more information.

Manually Configuring Drill Security

Before connecting to a data source, you can manually configure Drill security features and secure communication pathways to a secure Drill cluster. Security features that you can manually configure include:

- Authentication
  - Drillbit node (configure_server_auth.html)
  - Drill clients (drill_connectors.html)

- Impersonation
  - User impersonation (configure_user_impersonation.html)
  - User impersonation with Hive (configure_user_impersonation_hive.html)
- Authorization
  - Drill supports restricting an authenticated user's capabilities. See Configuring User Impersonation with Hive Authorization. (configure_user_impersonation_hive.html)
- Encryption
  - As of Drill 1.11 and MapR 6.0, Drill supports encryption using SSL and SASL (MapR security and Kerberos). See Using SSL/TLS for Encryption (config-ssl-encrypt.html#config-ssl-encrypt) and Configuring MapR Security (MapR-SASL) for Drill (configure_server_auth.html#configure_server_auth).

Roles and Privileges (../Drill/roles_and_privleges.html)

Configuring User Impersonation (../Drill/configure_user_impersonation.html)

Configuring User Impersonation with Hive (../Drill/configure_user_impersonation_hive.html)

Configuring Drill to Use Kerberos with Hive Metastore
(../Drill/configuring_drill_to_use_kerberos_with_hive_metastore.html)

Configuring Web Console and Web API Security (../Drill/ConfigWebConsolSecurity.html)

Configuring MapR Security (MapR-SASL) for Drill (../Drill/configure_server_auth.html)

MapR Drill supports authentication through the MapR-SASL mechanism and also supports encryption. Authentication is the process of establishing confidence of authenticity. Encryption is the process of converting information or data in plain text into ciphertext to prevent unauthorized access.

File ACEs on Drill Views (../Drill/file_aces_on_drill_views.html)

Using SSL/TLS for Encryption (../Drill/config-ssl-encrypt.html)

You can enable SSL for Drill in a secure or unsecure MapR cluster. SSL (Secure Sockets Layer), more recently called TLS, is a security mechanism that encrypts data passed between the Drill client and Drillbit (server). SSL also provides one-way authentication through which the Drill client verifies the identity of the Drillbit.