Enabling and Disabling Security Features on Your Cluster

Wire-level security encrypts data transmission between the nodes in your cluster.

⚠ Security features are turned *off* by default.

Enabling Wire-Level Security

When you set up a cluster, run the `configure.sh` script on each node that you want to add to the cluster. To enable security for the cluster, follow these steps in order:

1. If the cluster is running, shut it down.
2. Run the `configure.sh` script with the `-secure -genkeys` options on the first CLDB node in your cluster.

   ```
   /opt/mapr/server/configure.sh -N <cluster_name> -secure -genkeys -Z
   <Zookeeper_node_list> -C <CLDB_node_list>
   ```

   Where both `<Zookeeper_node_list>` and `<CLDB_node_list>` have the form `hostname[:port_no][,hostname[:port_no]...]`.

   ⚠ You **must run** `configure.sh -genkeys` *once* on one CLDB node, since the resulting files must be copied to other nodes.

   This command generates four files in the `/opt/mapr/conf` directory:
   - `cldb.key`
   - `maprserverticket`
   - `ssl_keystore`
   - `ssl_truststore`
3. Copy the `cldb.key` file to any node that has the CLDB or Zookeeper service installed.
4. Copy the `maprserverticket`, `ssl_keystore`, and `ssl_truststore` files to the `/opt/mapr/conf` directory of *every* node in the cluster.
5. Verify that the files from the previous step are owned by the user that runs cluster services. This user is `mapr` by default. Also, the `maprserverticket` and `ssl_keystore` files must have their UNIX permission-mode bits set to `600`, and the `ssl_truststore` file must be readable to all users.
6. Run `configure.sh -secure` on each existing node in the cluster. The `-secure` option indicates that the node is secure.

   ⚠ You must also do this on any nodes that you add to the cluster in the future.
7. Copy the `ssl_truststore` file to any client nodes outside the cluster.

   ⚠ If you run `configure.sh -secure` on a node *before* you copy the necessary files to that node, the command fails.
8. Log in as the mapr superuser using the maprlogin command:

   `maprlogin password` (in this command, `password` is literal text)
9. Run the `hadoop mfs -setnetworkencryption on <object>` command for every table, file, and directory object in MapR-FS whose traffic you wish to encrypt.

   ⚠

> ⚠ The network encryption setting is inherited by new objects. Once encryption is turned on for a directory, all new directories, files, and tables created under that directory are automatically encrypted.

10. If clients will connect to multiple secure clusters, merge the `ssl_truststore` files with the `/opt/mapr/server/manageSSLKeys.sh` tool. See Setting Up the Client for more information on MapR clients.

Generating Certificates After Initial Installation

When you run the `configure.sh` script at initial installation, but do not specify the `-genkeys` option, the script generates a `ssl_keystore` file for use by the web server for the MapR Control system. When the `configure.sh` script is run with the `-genkeys` option after initial installation, the system detects the existing `ssl_keystore` file and exits with an error to prevent inadvertent deletion or reuse of the `ssl_keystore` file. The error message will look similar to the following example:

```
/opt/mapr/server/configure.sh -secure -genkeys -C $CLDB_GRP -Z $ZK_GRP -RM $RM -HS
$HISTORYSERVER
<hostname1>: Configuring Hadoop-2.x at /opt/mapr/hadoop/hadoop-2.x
<hostname1>: Done configuring Hadoop
<hostname1>: CLDB node list: <hostname1>:7222,<hostname2>:7222,<hostname3>:7222
<hostname1>: Zookeeper node list: <hostname1>:5181,<hostname2>:5181,<hostname3>:5181
<hostname1>: Node setup configuration: cldb fileserver historyserver nfs nodemanager
resourcemanager webserver zookeeper
<hostname1>: Log can be found at: /opt/mapr/logs/configure.log
<hostname1>: /opt/mapr/conf/ssl_keystore already exists
<hostname1>: ERROR: could not generate ssl keys. See log file for more details
clush: <hostname1>: exited with exit code 1
```

- **On clusters without security features enabled**, the contents of the `ssl_keystore` file are unique to each node. In this case, manually delete the `ssl_keystore` file on each node, then run the command `configure.sh -genkeys`.
- **On clusters where you have customized the contents of the `ssl_keystore` file**, run the command `configure.sh -genkeys -nocerts` to preserve your customizations.

For general information on security tickets and certificates, see Tickets and Certificates.

System Behavior Changes After Enabling Security

After enabling security features for your cluster, the following behaviors change:

- Users must authenticate with the `maprlogin` utility.
- Components that have web UIs, such as the MapR Control System (MCS), Hive, and Oozie, require authentication.

  > ⚠ Note that you must also complete the PAM Configuration to set up user authentication for MCS logins.
- Several components that communicate over HTTP use HTTPS instead.
- Encryption is used for significant network traffic. Not all network traffic can be encrypted. Transmissions between ZooKeeper nodes are not encrypted.
- Access to a cluster using URIs that use the CLDB node's name or IP address, instead of the cluster name, is no longer supported, as in the following examples.
  - The following URIs no longer work after enabling security:
    `http://cldb1.cluster.com:7222/f1`
    `http://10.10.20.10:7221/f1`
  - The following URIs work after enabling security:
    `http:///f1 <access f1 in default cluster>`
    `http://my.cluster.com/f1`

In addition, several open source components require further configuration.

Disabling Wire-Level Security

To disable security features for your cluster:

1. If the cluster is running, shut it down.
2. Run the `configure.sh` script with the `-unsecure` option and specify the CLDB and ZooKeeper nodes.

```
configure.sh -unsecure -C <CLDB_Node> -Z <ZK_Node>
```

3. Start the cluster.

In `mapr-clusters.conf`, the cluster is changed from `secure=true` to `secure=false`.