MapR Tickets and PAM

Describes how to configure PAM.

The MapR Converged Data Platform supports Pluggable Authentication Modules (PAM) (PAM-Configuration.html#PAM-Configuration-MapRusesPluggableAut-d3e64) in the UNIX authentication stack. MapR provides a PAM Authenticator module that generates MapR tickets in conjunction with the `maprlogin` utility. After you install the MapR Converged Data Platform, the PAM Authenticator module is located at `*$INSTALL_DIR*/lib/libmapr_pam.so (SecurityArchitecture-AuthenticationArchitecture.html)` . Configuration files for PAM are located in the `/etc/pam.d` directory, and each UNIX operation, such as `su` , `login` , or `ssh` , has a specific PAM configuration file in that directory.

Configuring the PAM Authenticator on Ubuntu or SUSE

To configure the MapR PAM Authenticator, apend the following line to the end of the `/etc/pam.d/common-auth` file:

```
auth optional /opt/mapr/lib/libmapr_pam.so (SecurityArchitecture-AuthenticationArchitecture.html) # MapR PAM module
```

Warning: An absolute path to the location of the `libmapr_pam.so (SecurityArchitecture-AuthenticationArchitecture.html)` file is required. By default, this location is `$MAPR_HOME/lib/libmapr_pam.so (SecurityArchitecture-AuthenticationArchitecture.html)` .

Configuring the PAM Authenticator on Red Hat or CentOS

1. Insert the following line in the `/etc/pam.d/system-auth` file immediately before the first module that uses the `auth sufficient` configuration: `auth optional libmapr_pam.so (SecurityArchitecture-AuthenticationArchitecture.html) # MapR PAM module`
2. Append the string `try_first_pass` to the end of the module that uses `auth sufficient` , as in this example:
   Before modification:

   ```
   auth required pam_env.so
   auth sufficient pam_unix.so nullok
   auth requisite pam_succeed_if.so uid >= 500 quiet
   auth required pam_deny.so
   ```

   After modification, changes in **bold**:

   ```
   auth required pam_env.so
   auth optional libmapr_pam.so # MapR PAM module
   auth sufficient pam_unix.so nullok try_first_pass
   auth requisite pam_succeed_if.so uid >= 500 quiet
   auth required pam_deny.so
   ```

Configuring Debugging for PAM

To enable debugging for the client traffic used by the `maprlogin` utility, update the `/opt/mapr/conf/log4j.properties` file with the following line:

```
log4j.logger.com.mapr.login=DEBUG
```

After updating the `log4j.properties` file, trace the `com.mapr.login` package at the DEBUG level.

Be sure to update the correct instance of the `log4j.properties` file. Traffic specific to MapR, such as `maprlogin` and MapR Control System (MCS) traffic, uses the instance in the `/opt/mapr/conf` directory. Traffic used by the JobTracker, TaskTracker, and the `hadoop` command use the instance in the `/opt/mapr/hadoop/hadoop-<version>/conf` directory.

To perform the same tracing activity on the server side, modify the appropriate instance of the `log4j.properties` file on the server, or specify the page `com.mapr.login` in the MCS UI's tracing/logger settings. To trace PAM activity from the server, add the following line to the server's `log4j.properties` instance:

```
log4j.logger.net.sf.jpam=DEBUG
```

After modifying this setting, the server log will contain a message similar to the following:

```
2013-07-23 16:05:25,200 DEBUG Pam [1068409264@qtp-874242484-3]: Debug mode active.
```

Detailed information about PAM activity is written to the `/opt/mapr/logs/pam.log` file.

Other Packages

The following packages are not directly related to PAM, but can provide useful insights for subtler errors.

- `org.apache.hadoop.security` - This package contains Apache security code, including MapR enhancements. Tracing this package can provide information about what login configuration is in use.
- `com.mapr.fs.cldb.http.login` - This package contains code that the CLDB uses to validate `maprlogin` calls.

Common Issues

The Linux Documentation Project's HOWTO on LDAP Implementation has a section (http://www.tldp.org/HOWTO/archived/LDAP-Implementation-HOWTO/pamnss.html) on PAM and NSS that may prove helpful.

If a user's credentials appear valid, for example in a case where the `su` and `ssh` commands work normally, but PAM does not correctly authenticate, the issue may relate specifically to MapR's use of PAM as a normal user, compared to the usual case where PAM consumers run as the root user, causing permissions issues. The two most common issues relating to this condition are:

- The `/etc/shadow` directory is not readable to the `mapr` user. This directory is made readable to the `mapr` user during install, but some secure environments and configuration management tools undo these changes.
- A Kerberos PAM module is attempting to create and change the ownership of a kerberos ticket file. This attempt fails, since these changes require root privileges. Different Kerberos PAM modules can report errors differently, leading to difficulty tracking down root causes of errors. To address permissions problems with Kerberos PAM modules, configure the Kerberos PAM module to skip creating a ticket file, using the KDC only to validate the password. PAM configuration information is located in the `/etc/pam.d` directory. MapR can use a custom PAM configuration specified in the `web.conf` file.