Management Console Security

Management Console Security

SSL encrypted RMI (0.5 and above)

Current versions of the broker make use of SSL encryption to secure their RMI based JMX ConnectorServer for security purposes. This ships enabled by default, although the test SSL keystore used during development is not provided for security reasons (using this would provide no security as anyone could have access to it).

Broker Configuration

The broker configuration must be updated before the broker will start. This can be done either by disabling the SSL support, utilizing a purchased SSL certificate to create a keystore of your own, or using the example 'create-example-ssl-stores' script in the brokers bin/ directory to generate a self-signed keystore.

The broker must be configured with a keystore containing the private and public keys associated with its SSL certificate. This is accomplished by setting the Java environment properties *javax.net.ssl.keyStore* and *javax.net.ssl.keyStorePassword* respectively with the location and password of an appropriate SSL keystore. Entries for these properties exist in the brokers main configuration file alongside the other management settings (see below), although the command line options will still work and take precedence over the configuration file.

```
<management>
    <ssl>
        <enabled>true</enabled>
        <!-- Update below path to your keystore location, eg ${conf}/qpid.keystore  -->
        <keyStorePath>${prefix}/../test_resources/ssl/keystore.jks</keyStorePath>
        <keyStorePassword>password</keyStorePassword>
    </ssl>
</management>
```

JMX Management Console Configuration

If the broker makes use of an SSL certificate signed by a known signing CA (Certification Authority), the management console needs no extra configuration, and will make use of Java's built-in CA
truststore for certificate verification (you may however have to update the system-wide default truststore if your CA is not already present in it).

If however you wish to use a self-signed SSL certificate, then the management console must be provided with an SSL truststore containing a record for the SSL certificate so that it is able to validate it when presented by the broker. This is performed by setting the *javax.net.ssl.trustStore* and *javax.net.ssl.trustStorePassword* environment variables when starting the console. This can be done at the command line, or alternatively an example configuration has been made within the console's qpidmc.ini launcher configuration file that may pre-configured in advance for repeated usage. See the User Guide for more information on this configuration process.

JConsole Configuration

As with the JMX Management Console above, if the broker is using a self-signed SSL certificate then in order to connect remotely using JConsole, an appropriate trust store must be provided at startup. See JConsole for further details on configuration.

Additional Information

More information on Java's handling of SSL certificate verification and customizing the keystores can be found in the JSSE Reference Guide .

JMXMP (M4 and previous)

In previous releases of Qpid (M4 and below) the broker, can make use of Sun's Java Management Extensions Messaging Protocol (JMXMP) to provide encryption of the JMX connection, offering increased security over the default unencrypted RMI based JMX connection.

Download and Install

This is possible by adding the jmxremote_optional.jar as provided by Sun. This jar is covered by the Sun Binary Code License and is not compatible with the Apache License which is why this component is not bundled with Qpid.

Download the JMX Remote API 1.0.1_04 Reference Implementation from here. The included 'jmxremote-1_0_1-bin\lib\jmxremote_optional.jar' file must be added to the broker classpath:

First set your classpath to something like this:

```
CLASSPATH=jmxremote_optional.jar
```

Then, run qpid-server passing the following additional flag:

```
qpid-server -run:external-classpath=first
```

Following this the configuration option can be updated to enabled use of the JMXMP based JMXConnectorServer.

Broker Configuration

To enabled this security option change the *security-enabled* value in your broker configuration file.

```
     <management>
         <security-enabled>true</security-enabled>
     </management>
```

You may also (for M2 and earlier) need to set the following system properties using the environment variable QPID_OPTS:

QPID_OPTS="-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8999 -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false"

JMX Management Console Configuration

If you wish to connect to a broker configured to use JMXMP then the console also requires provision of the Optional sections of the JMX Remote API that are not included within the JavaSE platform.

In order to make it available to the console, place the 'jmxremote_optional.jar' (rename the file if any additional information is present in the file name) jar file within the 'plugins/jmxremote.sasl_1.0.1/' folder of the console release (on Mac OS X you will need to select 'Show package contents' from the context menu whilst selecting the management console bundle in order to reveal the inner file tree).

Following the the console will automatically load the JMX Remote Optional classes and attempt the JMXMP connection when connecting to a JMXMP enabled broker.

User Accounts & Access Rights

In order to access the management operations via JMX, users must have an account and have been assigned appropriate access rights.