10th Floor, 88 Wood Street
London
EC2V 7RS

Office:0845 868 5041
Fax:0870 471 2039
www.w3partnership.com

Connecting WebSphere MQ & Forum Sentry

AUTHOR: GRAHAM WOODS
W3PARTNERSHIP LTD

DATE: 30 APRIL 2013

Contents

1 Introduction

This document is intended to offer a step-by-step guide to implementing a connection between WebSphere MQ & a Forum Sentry device.

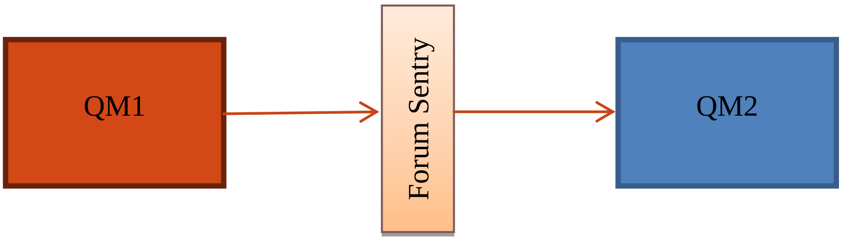
The guide initially details the steps to implement an unsecured connection. Subsequent steps are then detailed to secure the connection via TLS/SSL.

The product versions used in proving the statements used within the guide are:

Product	Version
WebSphere MQ	v7.1.0.0
Forum Sentry	v8.0.642
Operating System	Linux 2.6.18-274.3.1.el5

1.1 Overview

For the purposes of this document, the following topology is used.



Messages are sent from the WMQ queue manager, QM1, via a server connection channel (QM1_TO_FS) to the Forum Sentry device.

The messages are not transformed or enriched within the device. They are merely forwarded to the WMQ queue manager, QM2, via another server connection channel (FS_TO_QM2).

Messages placed within queue Q1, defined to queue manager QM1, will be “read” by the Forum Sentry device and written to queue Q2, defined to queue manager QM2.

2 WebSphere MQ Configuration

On each of the queue managers, QM1 & QM2, define a server connection (SVRCONN) channel & local queue as appropriate. The commands below will create the channels without any security; this process is described later in the document – once messages have been proven to flow from QM1 to QM2 via Forum Sentry.

2.1 Queue Manager QM1

Log on to the server as user=mqm (or a user within the mqm group). Issue the following commands:

```
runmqsc QM1

DEFINE CHANNEL(QM1_TO_FS) CHLTYPE(SVRCONN)

DEFINE QLOCAL(Q1)
```

2.2 Queue Manager QM2

Log on to the server as user=mqm (or a user within the mqm group). Issue the following commands:

```
runmqsc QM2

DEFINE CHANNEL(FS_TO_QM2) CHLTYPE(SVRCONN)

DEFINE QLOCAL(Q2)
```

3 Forum Sentry Configuration

Log on to the Forum Sentry Administration Console to perform the following steps.

3.1 Create MQ Listener Network Policy

For each connection to/from WMQ, a separate Network Policy is required. When MQ messages are sent to Forum Sentry from a Queue Manager, a "MQ Listener Policy" is required:

- Click: "Network Policies" (within Gateway > Gateway Policies) & click on "New".
- Ensure the "IBM Websphere MQ" radio button is enabled and click "Next".
- Enable "Listener" radio button and click "Next".
- Enter an appropriate name for the MQ Listener Policy (e.g. MQListenerPolicy-QM1_TO_FS).
- Enter the server hostname (or ip-address) of the server hosting the MQ Queue Manager and click "Next".
- Enter the port on which the MQ Queue Manager is listening and click "Next".
- Enter the Server Connection (SVRCONN) channel to connect with the MQ Queue Manager click "Next".
- Enter the MQ Queue Manager name and click "Next".
- Enter the MQ Queue that Forum Sentry is to read messages from and click "Next".
- Choose the MQ Message Format[1] – JMS Format or MQ Format and click "Next".
- If "JMS Format" was selected, enable either the "JMS Text Message" or the "JMS Map Message" and click "Next".
- If "JMS Map Message" was selected, enter the map field name and click "Next".
- Enter the MQ user (e.g. mqm) and click "Next".
- Enter the MQ user password and click "Next".
- Leave "Requires SSL" un-checked and click "Next".
- Leave "Require Authentication" un-checked and click "Next".
- Leave "Synchronous Policy" un-checked and click "Next".
- Leave "Template Name" with its default value and click "Finish".

3.2 Create MQ Remote Network Policy

For each connection to/from WMQ, a separate Network Policy is required. When MQ messages are sent from Forum Sentry to a Queue Manager, a "MQ Remote Policy" is required:

- Click: "Network Policies" (within Gateway > Gateway Policies) & click on "New".
- Ensure the "IBM Websphere MQ" radio button is enabled and click "Next".
- Enable "Remote" radio button and click "Next".
- Enter an appropriate name for the MQ Remote Policy (e.g. MQRemotePolicy-FS_TO_QM2).

[1] To determine this value, send a message to the queue and inspect the MQMD Message Format (perhaps using a tool such as RFHUtil).

- Enter the server hostname (or ip-address) of the server hosting the MQ Queue Manager and click "Next".
- Enter the port on which the MQ Queue Manager is listening and click "Next".
- Enter the Server Connection (SVRCONN) channel to connect with the MQ Queue Manager click "Next".
- Enter the MQ Queue Manager name and click "Next".
- Enter the MQ Queue that Forum Sentry is to write messages to and click "Next".
- Choose the appropriate delivery mode for the messages to the queue – persistent, non-persistent or as defined by the queue.
- Choose the MQ Message Format[2] – JMS Format or MQ Format and click "Next".
- If "JMS Format" was selected, enable either the "JMS Text Message" or the "JMS Map Message" and click "Next".
- If "JMS Map Message" was selected, enter the map field name and click "Next".
- Enter the MQ user (e.g. mqm) and click "Next".
- Enter the MQ user password and click "Next".
- Leave "Requires SSL" un-checked and click "Next".
- Leave "Synchronous Policy" un-checked and click "Next".
- Set "Time to Live (seconds)" to zero and click "Finish".

3.3 Create XML Policy within Forum Sentry

For each transfer of a message via Forum Sentry, a separate XML Policy is required. An XML Policy is essentially a set of rules that provide a policy for processing of an XML message through the system. The XML Policies created below perform neither transformation nor enrichment of the message.

- Click: "XML Policies" (within Gateway > Gateway Policies) & click on "New".
- Enter an appropriate name for the XML Policy (e.g. MQ_QM1_to_QM2_XML_Policy) and an appropriate description. Click on "Next".
- Ensure the radio button "Select from existing listener policies" is enabled and choose the appropriate MQ Listener Policy (i.e. MQListenerPolicy-QM1_TO_FS) from the drop-down.
- Ensure the radio button "Select from existing remote policies" is enabled and choose the appropriate MQ Remote Policy (i.e. MQRemotePolicy-FS_TO_QM2) from the drop-down.
- Click "Finish".

3.4 Test Connection

The system is now in a position to process a message. Put a message to queue Q1, on queue manager QM1 – i.e. via a utility such as RFHUtil, via WMQ Explorer or via the command line using the sample program *amqsput*.

The message should arrive unchanged on Q2.

Evidence of the message passing via Forum Sentry can be seen from the internal logs:

[2] To determine this value, send a message to the queue and inspect the MQMD Message Format (perhaps using RFHUtil).

- Click: “Internal Logs” (within Diagnostics > Logging) & click on “Today” within “System Logs”.
- The logs displayed should show evidence of the message being processed.

4 Addition of Security

Now messages are flowing between queue managers and Forum Sentry, we can add the additional steps to secure the communication.

The commands issued below create and use self-signed certificates. The appropriate additional steps will need to be performed to have certificates signed by the appropriate Certificate Authority.

In order to be able to issue any of the security commands detailed below, the security component of WebSphere MQ, called *MQSeriesGSKit*, must have been installed when the WMQ product was installed.

4.1 Overview: connecting a client to a queue manager securely

Secure communications that use the SSL (or TLS) cryptographic security protocols involve setting up the communication channels and managing the digital certificates that are used for authentication.

To set up your SSL/TLS installation, the appropriate channels must be defined to use SSL/TLS and digital certificates must be obtained and managed.

This collection of topics introduces the tasks involved in setting up SSL communications, and provides step-by-step guidance on completing those tasks.

During the SSL/TLS handshake, the SSL/TLS client always obtains and validates a digital certificate from the server. With the WMQ implementation, the SSL/TLS server always requests a certificate from the client.

WMQ uses the `ibmwebspheremq` prefix on a label to avoid confusion with certificates for other products. Ensure that you specify the entire certificate label in lowercase.

The SSL/TLS server always validates the client certificate if one is sent. If the client does not send a certificate, authentication fails only if the end of the channel that is acting as the SSL/TLS server is defined with either the SSLCAUTH parameter set to REQUIRED or an SSLPEER parameter value set.

4.2 Using self-signed certificates for mutual authentication of a client and queue manager

The instructions to implement mutual authentication between a client and a queue manager, by using self-signed SSL/TLS certificates, are detailed in the sections below. As a summary, the steps required are:

- Prepare a key repository on each client and queue manager.
- Create self-signed certificates for each client and queue manager:
- Extract a copy of each certificate as appropriate.
- Transfer the public part of the client certificate to the Queue Manager system and vice versa, using a utility such as FTP.
- Add the partner certificate to the key repository for each client and queue manager:
- Define a server-connection channel on the Queue Manager.

4.2.1 Setting up a key repository

An SSL/TLS connection requires a *key repository* at each end of the connection. Each WMQ queue manager and WMQ MQI client must have access to a key repository.

On Linux systems, digital certificates are stored in a key database file. These digital certificates have labels. A specific label associates a personal certificate with a queue manager or WMQ MQI client. SSL/TLS uses that certificate for authentication purposes. On Linux, WMQ uses *ibmwebspheremq* as a label prefix to avoid confusion with certificates for other products. The prefix is followed by the name of the queue manager or the WMQ MQI client user logon ID, changed to lowercase. The entire certificate label must be specified in lowercase.

Use the following command to create a new CMS key database file for either a queue manager or a WMQ MQI client:

runmqakm -keydb -create -db *filename* -pw *password* -type *cms* -stash -fips -strong

where:

-db *filename*

Specifies the fully qualified file name of a CMS key database, and must have a file extension of `.kdb`; use the value “key.kdb”

-pw *password*

Specifies the password for the CMS key database

-type *cms*

Specifies the type of database - for WMQ, it must be *cms*

-stash

Saves the key database password to a file.

-fips

Disables the use of the BSafe cryptographic library; only the ICC component is used and this component must be successfully initialized in FIPS mode. When in FIPS mode, the ICC component uses algorithms that are FIPS 140-2 validated. If the ICC component does not initialize in FIPS mode, the runmqakm command fails.

-strong

Checks that the password entered satisfies the minimum requirements for password strength; the minimum requirements for a password are as follows:

- The password must be a minimum length of 14 characters.
- The password must contain a minimum of one lowercase character, one uppercase character, and one digit or special character. Special characters include the asterisk (*), the dollar sign ($), the number sign (#), and the percent sign (%). A space is classified as a special character.
- Each character can occur a maximum of three times in a password.
- A maximum of two consecutive characters in the password can be identical.
- All characters are in the standard ASCII printable character set within the range 0x20 - 0x7E.

<u>Full command</u>

On both Queue Manager servers:

runmqakm -keydb -create -db key.kdb -pw *password*[3] -type *cms* -stash -fips –strong

[3] A suitable password must be chosen, and recorded as appropriate, by the Administrator performing these steps. Note: a strong password must be used – see description for –strong parameter

4.2.2 Accessing and securing your key database files

The key database files might not have appropriate access permissions.

For a queue manager, the permissions on the key database files must be set so that the queue manager and channel processes can read them when necessary, but other users cannot read or modify them. Normally, the mqm user needs read permission. If the key database file has been created by logging in as the mqm user, then the permissions are probably sufficient; if created using another user in the mqm group, grant read permissions to other users in the mqm group.

Similarly for a client, the permissions on the key database files must be set so that client application processes can read them when necessary, but other users cannot read or modify them. Normally, the user under which the client process runs needs read permission. If the key database file has been created by logging in as that user, then the permissions are probably sufficient; if they were created by another user in that group, grant read permissions to other users in the group.

Set the permissions on the files *key*.kdb, *key*.sth, *key*.crl, and *key*.rdb (where *key* is the stem name of the key database), to read and write for the file owner, and to read for the mqm or client user group (-rw-r-----).

4.2.3 Creating a self-signed personal certificate

Issue the following command to create the self-signed certificate:

runmqakm -cert -create -db *filename* -pw *password* -label *label* -dn *distinguished_name* -size *key_size* -x509version *version* -expire *days* -fips -sig_alg *algorithm*

where:

-db *filename*

Specifies the fully qualified file name of a CMS key database, and must have a file extension of `.kdb`.

-pw *password*

Specifies the password for the CMS key database.

-label *label*

The key label attached to the certificate

-dn *distinguished_name*

The X.500 distinguished name enclosed in double quotation marks. At least one attribute is required. Multiple OU or DC attributes can be supplied.

-size *key_size*

The key size; for runmqakm, the value can be 512, 1024, 2048 or 4096.

-x509version *version*

The version of X.509 certificate to create; the value can be 1, 2, or 3 (default=3).

-expire *days*

The number of days for which the certificate is valid - default is 365 days.

-fips

Disables the use of the BSafe cryptographic library; only the ICC component is used and this component must be successfully initialized in FIPS mode. When in FIPS mode, the ICC component uses algorithms that are FIPS 140-2 validated. If the ICC component does not initialize in FIPS mode, the runmqakm command fails.

-sig_alg ***algorithm***

The hashing algorithm used during the creation of a self-signed certificate. This hashing algorithm is used to create the signature associated with the newly created self-signed certificate.

The value can be: md5, MD5_WITH_RSA, MD5WithRSA, SHA_WITH_DSA, SHA_WITH_RSA, sha1, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, sha224, SHA224_WITH_RSA, SHA224WithDSA, SHA224WithECDSA, SHA224WithRSA, sha256, SHA256_WITH_RSA, SHA256WithDSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithRSA, sha384, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, sha512, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHAWithDSA, SHAWithRSA, EC_ecdsa_with_SHA1, EC_ecdsa_with_SHA224, EC_ecdsa_with_SHA256, EC_ecdsa_with_SHA384, or EC_ecdsa_with_SHA512.

The default value is SHA1WithRSA.

Full command

On both Queue Manager servers:

runmqakm -cert -create –db key.kdb –pw *password* –label ibmwebspheremq<*qmgr*> -dn "*distinguished name*" –size 2048 -x509version 3 –expire *days* –fips –sig_alg *signature algorithm*

4.2.4 Extracting the public part of a self-signed certificate from a key repository

Perform the following on the machine from which you want to extract the public part of a self-signed certificate:

runmqakm -cert -extract -db *filename* -pw *password* -label *label* -target *filename* -format *ascii* -fips

where:

-db ***filename***

Specifies the fully qualified file name of a CMS key database, and must have a file extension of `.kdb`.

-pw ***password***

Specifies the password for the CMS key database.

-label ***label***

The key label attached to the certificate

-target ***filename***

The name of the destination file.

-format ***ascii***

The format of the certificate; the value is *`ascii`* for Base64-encoded ASCII or `binary` for Binary DER data -the default is *`ascii`*.

-fips

Disables the of the BSafe cryptographic library; only the ICC component is used and this component must be successfully initialized in FIPS mode. When in FIPS mode, the ICC component uses algorithms that are FIPS 140-2 validated. If the ICC component does not initialize in FIPS mode, the runmqakm command fails.

Full command

runmqakm -cert -extract –db *keystore_filename* -pw *password* -label ibmwebspheremq<*qmgr*> -target *filename* -fips

4.2.5 Exchanging self-signed certificates

Exchange the certificates extracted in the section above; if FTP is used, ensure the correct format is used.

Note: the commands to date have assumed Base64 encoded X.509 certificates.

Transfer the following certificate types in *binary* format:

- DER encoded binary X.509
- PKCS #7 (CA certificates)
- PKCS #12 (personal certificates)

Transfer the following certificate types in ASCII format:

- PEM (privacy-enhanced mail)
- Base64 encoded X.509

4.2.6 Adding the public part of a self-signed certificate into a key repository

Once the public part of the certificate has been extracted and moved to the opposing server, perform the following steps to add it to the key repository:

runmqakm -cert -add -db *filename* -pw *password* -label *label* -file *filename -*format *ascii* -fips

where:

-db *filename*

Specifies the fully qualified file name of a CMS key database, and must have a file extension of `.kdb`.

-pw *password*

Specifies the password for the CMS key database.

-label *label*

The key label attached to the certificate.

-file *filename*

The name of the file containing the extracted certificate.

-format *ascii*

The format of the certificate; the value is *`ascii`* for Base64-encoded ASCII or `binary` for Binary DER data -the default is *`ascii`*.

-fips

Disables the of the BSafe cryptographic library; only the ICC component is used and this component must be successfully initialized in FIPS mode. When in FIPS mode, the ICC component uses algorithms that are FIPS 140-2 validated. If the ICC component does not initialize in FIPS mode, the runmqakm command fails.

Full command

runmqakm -cert -add -db key.kdb -pw *password* -label ibmwebspheremq<*qmgr*> -file *filename* -format ascii -fips

4.2.7 MQ Object Changes

Amend the definition of each server-connection channel already defined to each queue manager. The queue manager definition requires amendment also. On QM1:

```
runmqsc QM1

ALTER QMGR SSLFIPS (YES)

ALTER CHANNEL(QM1_TO_FS) CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)

STOP CHANNEL(QM1_TO_FS)

START CHANNEL(QM1_TO_FS)
```

And for QM2:

```
runmqsc QM2

ALTER QMGR SSLFIPS (YES)

ALTER CHANNEL(FS_TO_QM2) CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)

STOP CHANNEL(FS_TO_QM2)

START CHANNEL(FS_TO_QM2)
```

4.2.8 Set SSLPEER on MQ Channel

Once the secure connection has been proven, the SSLPEER value can be set on the Server Connection channel. This ensures that only messages with a specific DN are accepted. To set the channel SSLPEER value, perform the following:

- Ensure a test message has been sent from the client to the MQ Queue Manager once the steps above have been carried out.
- On the MQ server, as user=mqm, issue the following commands:
- runmqsc <QueueManagerName>
    - dis chs(<ServerConnctionChannelName>) all

    This will display the SSLPEER property of the channel status. This <value> now needs to be added to the channel definition itself:

    - alter chl(<ServerConnctionChannelName>) SSLPEER(<value>)
    - stop chl(<ServerConnctionChannelName>)
    - start chl(<ServerConnctionChannelName>)

4.3 Forum Sentry Configuration

The sections below define the steps required to configure the Forum Sentry communication. It is assumed that the user has access to the Forum Sentry Administration Console.

4.3.1 Define Forum Sentry SSL Key Pair

In order to create the internal Forum Sentry PKCS12 key pair, perform the following:

- Click: “Keys” (within Resources > PKI)
- Choose: “New”
- Ensure “PKCS Key Pair” radio button is enabled. Click: “Next”.
- Enter a name for the key – e.g. FS_Key_<YYYYMMDD>
- Choose: “RSA” for the Algorithm & “2048” for the Key Size.
- Enter some random data to Seed Entry & click “Next”.
- Enter the following values as the Identifying Information:

o	Common Name:	<Enter an appropriate value>
o	Email Address:	<Enter an appropriate value>
o	Organizational Unit(s):	<Enter an appropriate value>
o	Organizational Name:	<Enter an appropriate value>
o	City:	<Enter an appropriate value>
o	State/Province:	<Enter an appropriate value>
o	Country Code:	<Enter an appropriate value>

- Ensure the radio button to generate a self-signed certificate is enabled and enter a period of 730 days.
- Click: “Next” and the Key Details will now be displayed.

Now, perform the next section to extract the public key.

4.3.2 Extract Forum Sentry Public Key

Following on from the previous section, perform the following to extract the Public Key of the generated PKCS12 Key Pair. This key will be imported to the WMQ key stores you have defined above.

- Click: “Keys” (within Resources > PKI) & click on the key (not the associated certificate) that you’ve just created.
- Click on “PEM” & “Save File”.
- Double-click on the saved file to open it.
- Click on the “Details” tab and choose: “Copy to File….”.
- Click: “Next” & enable the “Base-64 encoded X.509” radio button.
- Click: “Next” & provide a directory & file name for the exported key – e.g. FS_Key_<YYYYMMDD>.
- Click: “Next” & then “Finish”.

4.3.3 Import WMQ Public Key

To import an extracted WMQ queue manager public key, perform the steps below. This should be done for all public keys that are to be added to the Forum Sentry key store – i.e. the public keys extracted on both servers hosting queue managers QM1 & QM2.

- Click: "Keys" (within Resources > PKI) & click on "Import".
- Ensure the "X.509 or PKCS#7 Public Certificates" radio button is enabled and click on "Next".
- Ensure "File upload" radio button is enabled and click on "Next".
- Enter the name of the key (e.g. QM1_PublicKey, QM2_PublicKey) & click on "Browse" in order to locate the actual key file (previously copied to your desktop). Make sure "Create Signer Group from Certificate Chain" is enabled & click "Submit".

4.3.4 Create Forum Sentry Security Policy

For each connection to/from WMQ, a separate Security Initiation Policy[4] is required. Perform the following steps:

- Click: "SSL" (within Resources > Security Policies) & click on "New".
- Ensure the "Initiation" radio button is enabled and click "Next".
- Give the policy an appropriate name – i.e. QM1toFS_SSL_Initiation_Policy.
- From the drop-down for "Authenticate to Remote Server using Key Pair", select the PKCS12 key pair generated in section 4.3.1.
- From the drop-down for "Authenticate the Remote Server using Signer Group[5]", select the appropriate signer group that was auto-defined when importing the public key (e.g. QM1_PublicKey) – see section above.
- Tick the check-box for "Ignore Server Hostname Verification" and leave the TLSv1 & SSLv3 check boxes checked and click on "Create".

4.3.5 Amend MQ Listener Network Policy within Forum Sentry

The MQ Listener Policy already defined must be amended to use SSL:

- Click: "Network Policies" (within Gateway > Gateway Policies) & click on the policy.
- Click on "Requires SSL" and tick the check-box and click "Next".
- Select the appropriate SSL Initiation policy from the drop-down, and click "Next".
- Select the appropriate SSL Cipher Spec (as being used by the MQ SVRCONN channel).
- Leave "Require Authentication" un-checked and click "Next".
- Leave "Synchronous Policy" un-checked and click "Next".
- Leave "Template Name" with its default value and click "Finish".

[4] A WMQ queue manager is regarded as a client connection in all cases – regardless of the message direction. Hence, SSL Initiation Policies are created rather than SSL Termination Policies.

[5] Once the initiation policy has been created, it can be simplified by setting "Authenticate to Remote Server using Key Pair" to "Authentication not required" & "Ignore Server Hostname Verification" to "Do not authenticate". Once the connection over SSL with no authentication has been proven, the original values should be re-instated and the connection attempted again. Note, when testing without authentication, the WMQ Server Connection (SVRCONN) channel must have its SSL parameters SSLCAUTH & SSLCIPH set to "OPTIONAL" & no value respectively.

4.3.6 Create MQ Remote Network Policy within Forum Sentry

The MQ Remote Policy must be amended to use SSL:

- Click: “Network Policies” (within Gateway > Gateway Policies) & click on the policy.
- Click on “Requires SSL” and tick the check-box and click “Next”.
- Select the appropriate SSL Initiation policy from the drop-down, and click “Next”.
- Select the appropriate SSL Cipher Spec (as being used by the MQ SVRCONN channel).
- Leave “Synchronous Policy” un-checked and click “Next”.
- Set “Time to Live (seconds)” to zero and click “Finish”.

4.3.7 Test the Connection

Repeat the steps outlined in section 3.4