Installation, Configuration and Basic Test of WebSphere MQ Advanced Message Security 7.5 in Linux

IBM Techdoc: 7041465

http://www-01.ibm.com/support/docview.wss?uid=swg27041465

Date last updated: 28-May-2014

Angel Rivera – rivera@us.ibm.com
IBM WebSphere MQ Support

+++ Objective

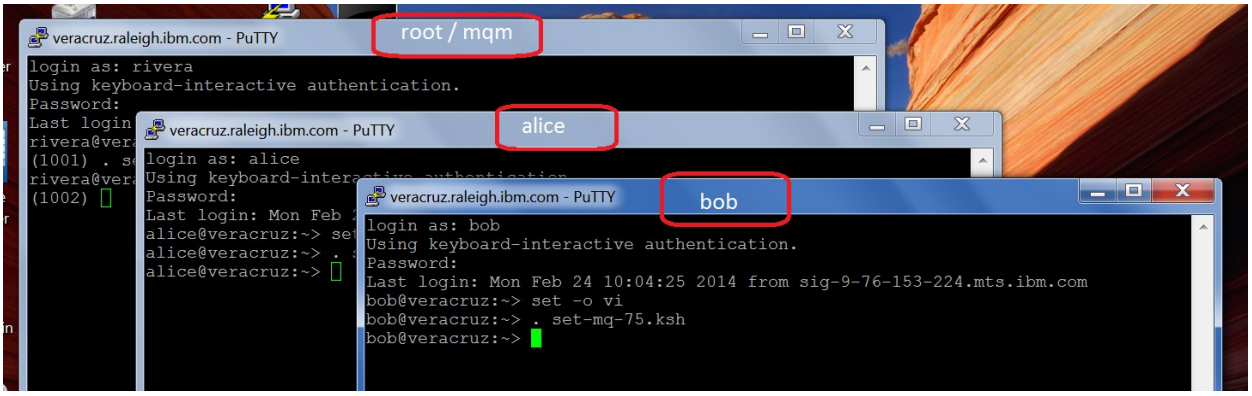
The objective of this technical document is to describe in detail how to install and configure for first usage the WebSphere MQ Advanced Message Security (AMS) version 7.5 in Linux. It also shows how to perform a basic test using the amqsput and amqsget samples.

WebSphere MQ provides transport-level security with features such as SSL/TLS over channels. However, by default, WebSphere MQ does not provide a method to encrypt and secure access to messages while they are at rest on queues.
If AMS is used in a WebSphere MQ environment, it is now possible to implement full end-to-end security.

The chapters in this techdoc are:

Because this scenario describes the tasks done by multiple users, it is best to create at least three (3) separate command prompt windows, which helps to reduce confusion.

Window 1: for users “root” and “mqm”
Window 2: for user “alice”
Window 3: for user “bob”



The material in this techdoc is based on the following chapter:

Knowledge Center:
http://www.ibm.com/support/knowledgecenter/SSFKSJ_7.5.0/com.ibm.mq.sec.doc/q014700_.htm

WebSphere MQ
> Security
> WebSphere MQ Advanced Message Security
> WebSphere MQ AMS overview
> User scenarios
IBM WebSphere MQ, Version 7.5 Operating Systems: UNIX, Linux, Windows
Quick Start Guide for UNIX platforms

+++ Update in May 2014:
The MQ Knowledge Center will replace the MQ Information Center.
Thus, the links for the Knowledge Center are now included.

+++ Related information

MQ 7.5 includes new attributes and system queues related to AMS. They are briefly described in the following references.

http://www.ibm.com/support/docview.wss?rs=171&uid=swg21578742
New MQ 7.1 and MQ 7.5 attributes in DISPLAY QMGR for new or migrated queue managers

Attribute SPLCAP (New in 7.5):
http://www.ibm.com/support/knowledgecenter/SSFKSJ_7.5.0/com.ibm.mq.ref.dev.doc/q102820_.htm
IBM WebSphere MQ Knowledge Center, Version 7.5 Operating Systems: UNIX, Linux, Windows
WebSphere MQ > Reference > Developing applications > MQI applications reference > Attributes of objects > Attributes for the queue manager
SPLCAP
Indicates whether security capabilities of WebSphere MQ Advanced Message Security are available for a queue manager.
MQCAP_SUPPORTED: This is the default value if the WebSphere MQ AMS component is installed for the installation that the queue manager is running under.
MQCAP_NOT_SUPPORTED

http://www-01.ibm.com/support/docview.wss?uid=swg21608033
New SYSTEM queues added in WebSphere MQ 7.1 and 7.5

SYSTEM.PROTECTION.ERROR.QUEUE
SYSTEM.PROTECTION.POLICY.QUEUE
These 2 queues are used by WebSphere MQ Advanced Message Security (AMS)
http://www.ibm.com/support/knowledgecenter/SSFKSJ_7.5.0/com.ibm.mq.sec.doc/q014700_.htm
WebSphere MQ > Security > WebSphere MQ Advanced Message Security > WebSphere MQ AMS overview > User Scenarios
Quick Start Guide for UNIX platforms
The queue SYSTEM.PROTECTION.ERROR.QUEUE is used to put error messages generated by the AMS code.
The queue SYSTEM.PROTECTION.POLICY.QUEUE stores the policies defined by the MQ Administration for protecting other queues with the AMS function.

++
+++ Chapter 1: Installing the AMS code
++

This chapter describes the installation of the AMS components.
You also need to install the MQ samples, which include amqsput and amqsget.

Use Window 1 and log in as root.

In WMQ 7.0, the Advanced Message Security (AMS) product was shipped and installed separately.

Starting with WMQ 7.5, the AMS code has been incorporated into the main product and the AMS code is now obtained with the download images from the IBM Passport Advantage site.

The information regarding AMS is now included with the MQ 7.5 Knowledge Center (instead of having a separate Information Center, like with AMS):
http://www.ibm.com/support/knowledgecenter/SSFKSJ_7.5.0/

Furthermore, in MQ 7.5 the installation of the AMS components is done in the same manner as the other MQ components.

In MQ AMS 7.5 in Linux, the filesets for AMS are packaged with the MQ server filesets.

You need to log in as user “root” to install the MQ filesets.

The following free redbook has an overview of the installation steps.
http://www.redbooks.ibm.com/redpieces/abstracts/sg248087.html?Open
WebSphere MQ V7.1 and V7.5 Features and Enhancements

Specifically the section:
Section 16.1 (Page 232) WebSphere MQ Advanced Message Security installation

The following names of the AMS packages on UNIX and Linux are used:
AIX: mqm.ams.rte
HP-UX: MQSERIES.MQM-AMS
Linux: MQSeriesAMS
Solaris: mqams

The techdoc mentioned below describes how to install MQ 7.5, including AMS.
In the particular test described in the document, this file set was used:
MQSeriesAMS_2-7.5.0-0.i386.rpm

Also, the techdoc describes how to created shell scripts to help to establish the proper environment for each version of MQ to use (see Chapter 2).
In this case the shell script to be used is: set-mq-75.ksh

Techdoc 7036779
http://www.ibm.com/support/docview.wss?rs=171&uid=swg27036779
Installing WebSphere MQ 7.5 to coexist with MQ 7.0.1 and MQ 7.1 in Linux
This techdoc shows all the steps to install WebSphere MQ 7.5 in Linux, while co-existing ("side-by-side") with MQ 7.0.1 and MQ 7.1.
The overall instructions apply also to other UNIX distributions: AIX, HP-UX and Solaris.
The chapters are:
Chapter 1: Installing MQ 7.5 side-by-side to MQ 7.0.1.9 and 7.1
Chapter 2: Need to run setmqenv to use MQ 7.5 commands
Chapter 3: Creating a queue manager under 7.5
Chapter 4: Remote access to the new MQ 7.5 queue manager
Chapter 5: Using MQ Explorer 7.5
Chapter 6: Migrating an MQ 7.0 queue manager to MQ 7.5
Chapter 7: Installing fix pack MQ 7.5.0.3

Note: This is the contents of the shell script set-mq-75.ksh

```
#!/usr/bin/ksh
# Name: set-mq-75.ksh
# Purpose: to setup the environment to run MQ 7.5
. /opt/mqm75/bin/setmqenv -n Installation2
# Additional MQ 7 directories for the PATH
export PATH=$PATH:$MQ_INSTALLATION_PATH/bin:
$MQ_INSTALLATION_PATH/java/bin:$MQ_INSTALLATION_PATH/samp/bin:
$MQ_INSTALLATION_PATH/samp/jms/samples:
# Add local directory for running Java/JMS programs
export CLASSPATH=$CLASSPATH:.
# end
```

++
+++ Chapter 2: Creating a queue manager and a queue
++

UNIX host: Linux SLES 11, x86 32-bit
WebSphere MQ 7.5.0.3

++ Example of the line commands to create a queue manager

Use Window 1 and log in as user “mqm”.

You need to log in as user “mqm” or a member of the MQ Administration group (group “mqm”).

- Create the queue manager.
The -u flag indicates which queue is going to be the dead letter queue.
Hint: Many MQ Explorer users hide the SYSTEM* queues and thus, if you use as the dlq the SYSTEM.DEAD.LETTER.QUEUE, then it will be hidden and you might not notice if there are messages in the dead letter queue

crtmqm -u DLQ QM_VERIFY_AMS

- Start the queue manager
strmqm QM_VERIFY_AMS

- Configure the queue manager
runmqsc QM_VERIFY_AMS

Define the testing queue which will be protected by AMS
define qlocal(TEST.Q)

Define a normal queue which will NOT be protected by AMS
define qlocal(Q1)

Define a listener. It is a good idea to specify the port number in the name in that way a quick look at the list of listeners will tell you the port number right away.
The default port is 1414, however here the port 1444 will be used instead.
define listener(LISTENER.1444) trptype(tcp) control(qmgr) port(1444)
start listener(LISTENER.1444)

Define a channel to be used by a remote MQ Explorer
define channel(SYSTEM.ADMIN.SVRCONN) chltype(SVRCONN)

```
## Define the DLQ
  define qlocal(DLQ) like(SYSTEM.DEAD.LETTER.QUEUE)

## For MQ 7.1 and 7.5 and if desiring to allow remote connections by an MQ
Administrator, to avoid return code 2035:
  set CHLAUTH(*) TYPE(BLOCKUSER) USERLIST('nobody','*MQADMIN')
  set CHLAUTH(SYSTEM.ADMIN.*) TYPE(BLOCKUSER) USERLIST('nobody')

  ## Display the attribute SPLCAP which shows if AMS is enabled
 display qmgr SPLCAP
 AMQ8408: Display Queue Manager details.
  QMNAME(QM_VERIFY_AMS)                SPLCAP(ENABLED)

  ## Display the 2 system queues used by AMS

 display ql(SYSTEM.PROTECTION*)
 AMQ8409: Display Queue details.
   QUEUE(SYSTEM.PROTECTION.ERROR.QUEUE)    TYPE(QLOCAL)
 AMQ8409: Display Queue details.
   QUEUE(SYSTEM.PROTECTION.POLICY.QUEUE)   TYPE(QLOCAL)


  ## exit runmqsc

  end
```

For more details on the channel authority records introduced in MQ 7.1see:

http://www.ibm.com/support/knowledgecenter/SSFKSJ_7.5.0/com.ibm.wmqfte.doc/mq_chlauth.htm
WebSphere MQ Managed File Transfer > Security
Connecting to a WebSphere MQ V7.1 or later queue manager in client mode with channel authentication
WebSphere® MQ V7.1 introduced channel authentication records to control more precisely access at a channel level. This change in behavior means that by default newly created WebSphere MQ V7.1 or later queue managers reject client connections from the Managed File Transfer component.

http://www.ibm.com/support/docview.wss?uid=swg21577137
WMQ 7.1 / 7.5 queue manager RC 2035 MQRC_NOT_AUTHORIZED or AMQ4036 when using client connection as an MQ Administrator

++
+++ Chapter 3: Creating and authorizing users
++

++ Creating users

+ Window 1: User root

Log in as user “root”.

Use line commands (useradd) or the YAST GUI or another administrative tool to create the following users:

alice
bob
fulano

Notice that the user “fulano” will be used in the last chapter, to show what happens when an unauthorized user tries to browse the AMS protected messages.

For the scenarios described in this document, these users are NOT MQ administrators, therefore, they should NOT belong to the group “mqm”. Remember that in UNIX, any member of the group “mqm” (either as primary or a set of groups), is automatically an MQ administrator.

In this scenario the users are members of the group “mqusers”.

id alice
uid=1008(alice) gid=1005(mqusers) groups=1005(mqusers)

id bob
uid=1009(bob) gid=1005(mqusers) groups=1005(mqusers)

id fulano
uid=1006(fulano) gid=1005(mqusers) groups=1005(mqusers)

++ Authorizing users

+ Window 1: User mqm

Log in as user “mqm”

Because this test machine has 3 versions of MQ (7.0.1, 7.1 and 7.5), it is necessary to run the setmqenv command to select the desired version of MQ to use.
For more details, see the techdoc 7036779 mentioned at the end of Chapter 1.
In this case, the shell script is for the MQ 7.5 runtime environment:

. set-mq-75.ksh

The following commands were used to authorize the users to connect to the queue manager and to work with the queue TEST.Q:

setmqaut -m QM_VERIFY_AMS -t qmgr -p alice -p bob +connect +inq +dsp
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p alice +put +browse +dsp
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p bob +get +browse +dsp

The following commands are for the advanced testing done in the last chapter:

setmqaut -m QM_VERIFY_AMS -t qmgr -p fulano +connect +inq +dsp
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p fulano +put +browse +dsp
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p fulano +get +browse +dsp

Note:
Technically speaking, the authority in MQ is based on the group membership of the user. Thus, the setmqaut command for user alice actually has the side effect of giving authority to ALL the users who belong to the same primary group as alice, that is 'mqusers'. This means that users bob and fulano will automatically be authorized similar to alice. This is equivalent to use the -g flag (for group) in setmqaut.

Also, it is necessary to allow the two users to browse the AMS system policy queue and put messages on the AMS error queue.

setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p alice -p bob +browse

setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p alice -p bob +put

++ Verification that users alice and bob can put/get messages from TEST.Q

Before proceeding with the AMS example, let's the amqsput and amqsget samples to verify that the users can put and get messages:

+ Window 2: User alice

Log in as user “alice”

Select to work with the MQ 7.5 environment:
. set-mq-75.ksh

Put a message

amqsput TEST.Q QM_VERIFY_AMS
Sample AMQSPUT0 start
target queue is TEST.Q
test-AMS

Sample AMQSPUT0 end

+ Window 3: User bob

Log in as user “bob”

Select to work with the MQ 7.5 environment:
. set-mq-75.ksh

Get a message

amqsget TEST.Q QM_VERIFY_AMS
Sample AMQSGET0 start
message <test-AMS>
no more messages
Sample AMQSGET0 end

++
+++ Chapter 4: Creating key database and certificates
++

To encrypt the message, the AMS interceptors require the public key of the sending users. Thus, the key database of user identities mapped to public and private keys must be created.

In this scenario, we are using self-signed certificate which can be created without using a Certificate Authority. For production systems, it is advisable not to use self-signed certificates however instead rely on certificates signed by a Certificate Authority.

+ Window 2: User alice

This is the window where you have already log in as alice

The umask used in this example is:

umask
0022

Create a new key database for user alice
The -p flag will create intermediate directories, if they do not yet exist. It is useful when dealing a deep directory tree.

mkdir /home/alice/.mqs -p

runmqakm -keydb -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -stash

The following are the directories and files that were created:

ls -dl /home/alice/.mqs
drwxr-xr-x 2 alice mqusers 4096 2014-02-24 06:27 /home/alice/.mqs

ls -l /home/alice/.mqs
-rw------- 1 alice mqusers 88 2014-02-24 06:27 alicekey.crl
-rw------- 1 alice mqusers 88 2014-02-24 06:27 alicekey.kdb
-rw------- 1 alice mqusers 88 2014-02-24 06:27 alicekey.rdb
-rw------- 1 alice mqusers 129 2014-02-24 06:27 alicekey.sth

Create a self-signed certificate identifying the user alice for use in encryption

runmqakm -cert -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert -dn "cn=alice,o=IBM,c=GB" -default_cert yes

Notes:
- The 'label' parameter specifies the name for the certificate, which interceptors will look up to receive necessary information.
- The 'DN' parameter specifies the details of the Distinguished Name (DN), which must be unique for each user.

Notice the increase in size for alicekey.kdb, which indicates that the new certificate is stored in that file.

ls -l /home/alice/.mqs
-rw------- 1 alice mqusers 88 2014-02-24 06:27 alicekey.crl
-rw------- 1 alice mqusers 5088 2014-02-24 06:30 alicekey.kdb (notice size increase)
-rw------- 1 alice mqusers 88 2014-02-24 06:27 alicekey.rdb
-rw------- 1 alice mqusers 129 2014-02-24 06:27 alicekey.sth

+ Window 3: User bob

This is the window where you have already log in as bob

The umask used in this example is:

umask
0022

Create a new key database for the user bob

mkdir /home/bob/.mqs -p

runmqakm -keydb -create -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -stash

The following are the directories and files that were created:

ls -dl /home/bob/.mqs
drwxr-xr-x 2 bob mqusers 4096 2014-02-24 06:36 /home/bob/.mqs

ls -l /home/bob/.mqs
-rw------- 1 bob mqusers 88 2014-02-24 06:36 bobkey.crl
-rw------- 1 bob mqusers 88 2014-02-24 06:36 bobkey.kdb
-rw------- 1 bob mqusers 88 2014-02-24 06:36 bobkey.rdb

-rw------- 1 bob mqusers 129 2014-02-24 06:36 bobkey.sth

Create a certificate identifying the user bob for use in encryption

runmqakm -cert -create -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Bob_Cert -dn "cn=bob,o=IBM,c=GB" -default_cert yes

ls -l /home/bob/.mqs
-rw------- 1 bob mqusers 88 2014-02-24 06:36 bobkey.crl
-rw------- 1 bob mqusers 5088 2014-02-24 06:37 bobkey.kdb (notice size increase)
-rw------- 1 bob mqusers 88 2014-02-24 06:36 bobkey.rdb
-rw------- 1 bob mqusers 129 2014-02-24 06:36 bobkey.sth

++
+++ Chapter 5: Creating keystore.conf
++

You must point WebSphere MQ Advanced Message Security interceptors to the directory where the key databases and certificates are located. This is done via the keystore.conf file, which hold that information in the plain text form.

Each user must have a separate keystore.conf file. This step should be done for both alice and bob.

The content of keystore.conf must be of the form:

cms.keystore = <dir>/keystore_file
cms.certificate = certificate_label

Notes:
- The path to the keystore file must be provided with no file extension.
- HOME/.mqs/keystore.conf is the default location where WebSphere MQ Advanced Message Security searches for the keystore.conf file.

+ Window 2: User alice

Create file:
vi /home/alice/.mqs/keystore.conf

The contents is:

cms.keystore = /home/alice/.mqs/alicekey
cms.certificate = Alice_Cert

ls -l .mqs
-rw------- 1 alice mqusers 88 2014-02-24 06:27 alicekey.crl
-rw------- 1 alice mqusers 5088 2014-02-24 06:30 alicekey.kdb
-rw------- 1 alice mqusers 88 2014-02-24 06:27 alicekey.rdb
-rw------- 1 alice mqusers 129 2014-02-24 06:27 alicekey.sth
-rw-r--r-- 1 alice mqusers 71 2014-02-24 06:43 keystore.conf (new file)

+ Window 3: User bob

Create file:
vi /home/bob/.mqs/keystore.conf

The contents is:

cms.keystore = /home/bob/.mqs/bobkey
cms.certificate = Bob_Cert

ls -l /home/bob/.mqs
-rw------- 1 bob mqusers 88 2014-02-24 06:36 bobkey.crl
-rw------- 1 bob mqusers 5088 2014-02-24 06:37 bobkey.kdb
-rw------- 1 bob mqusers 88 2014-02-24 06:36 bobkey.rdb
-rw------- 1 bob mqusers 129 2014-02-24 06:36 bobkey.sth
-rw-r--r-- 1 bob mqusers 65 2014-02-24 06:44 keystore.conf (new file)

++
+++ Chapter 6: Sharing Certificates
++

It is necessary to share the certificates between the two key databases so that each user can successfully identify each other.

Because the users are located in the same host, the directory /tmp will be used as the neutral directory to exchange the certificates between the users.

But if they were located in different boxes, then you will need to use ftp and specify the file transfer as “ascii”.

+ Window 2: User alice

Export the certificate identifying alice to a file located in /tmp.
The resulting file will be written as ascii text, which is the default (-format ascii).

runmqakm -cert -extract -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert -target /tmp/alice_public.arm

Allow the certificate to be read by others

chmod 644 /tmp/alice_public.arm

ls -l /tmp/*.arm
-rw-r--r-- 1 alice mqusers 692 2014-02-24 06:55 /tmp/alice_public.arm

+ Window 3: User bob

Add the certificate from alice into bob's keystore:

runmqakm -cert -add -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert -file /tmp/alice_public.arm

ls -l /home/bob/.mqs
-rw------- 1 bob mqusers 88 2014-02-24 06:36 bobkey.crl
-rw------- 1 bob mqusers 10088 2014-02-24 06:56 bobkey.kdb (notice size increase)
-rw------- 1 bob mqusers 88 2014-02-24 06:36 bobkey.rdb
-rw------- 1 bob mqusers 129 2014-02-24 06:36 bobkey.sth
-rw-r--r-- 1 bob mqusers 65 2014-02-24 06:44 keystore.conf

Print the details of the certificate for alice, to verify that it is indeed in the keystore,
runmqakm -cert -details -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert

+ begin excerpt

Label : Alice_Cert
Key Size : 1024
Version : X509 V3
Serial : 6e090f5404c54ddb
Issuer : CN=alice,O=IBM,C=GB
Subject : CN=alice,O=IBM,C=GB
Not Before : February 23, 2014 6:30:50 AM EST
Not After : February 24, 2015 6:30:50 AM EST
Public Key
   30 81 9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01
...
Value
   5D D0 6A 65 05 C3 27 8D 10 EF FB FB 58 3F 78 F7
...
Trust Status : Enabled

+ end excerpt

Export the certificate identifying bob to a file located in /tmp:

runmqakm -cert -extract -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Bob_Cert -target /tmp/bob_public.arm

Allow the certificate to be read by others

chmod 644 /tmp/bob_public.arm

ls -l /tmp/*.arm
-rw-r--r-- 1 alice mqusers 692 2014-02-24 06:55 /tmp/alice_public.arm
-rw-r--r-- 1 bob mqusers 684 2014-02-24 06:58 /tmp/bob_public.arm

+ Window 2: User alice

Add the certificate for bob to alice's keystore:

runmqakm -cert -add -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert -file /tmp/bob_public.arm

ls -l /home/alice/.mqs

```
-rw------- 1 alice mqusers    88 2014-02-24 06:27 alicekey.crl
-rw------- 1 alice mqusers 10088 2014-02-24 07:00 alicekey.kdb (notice size increase)
-rw------- 1 alice mqusers    88 2014-02-24 06:27 alicekey.rdb
-rw------- 1 alice mqusers   129 2014-02-24 06:27 alicekey.sth
-rw-r--r-- 1 alice mqusers    71 2014-02-24 06:43 keystore.conf
```

Print the details

runmqakm -cert -details -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert

(Similar results as for alice)

++
+++ Chapter 7: Defining queue policy
++

Let's define protection policies using the “setmqspl” command.

Each policy name must be the same as the queue name it is to be applied to.

+ Window 1: User mqm

Example
This is an example of a policy defined for the TEST.Q queue.
The messages are signed by the user alice using the SHA1 algorithm, and encrypted using the 256-bit AES algorithm.
The user alice is the only valid sender and the user bob is the only receiver of the messages on this queue:

setmqspl -m QM_VERIFY_AMS -p TEST.Q -s SHA1 -a "CN=alice,O=IBM,C=GB" -e AES256 -r "CN=bob,O=IBM,C=GB"

Note: The DNs need to match exactly those specified in the receptive user's certificate from the key database.

Verify the policy:

dspmqspl -m QM_VERIFY_AMS
Policy Details:
Policy name: TEST.Q
Quality of protection: PRIVACY
Signature algorithm: SHA1
Encryption algorithm: AES256
Signer DNs:
  CN=alice,O=IBM,C=GB
Recipient DNs:
  CN=bob,O=IBM,C=GB
Toleration: 0

Best Practice:

The command “dmpmqcfg” was introduced in MQ 7.1 to provide a way to export the object definitions and authorizations of a queue manager, which could be used as backup or as a way to create a new queue manager with the same object definitions and authorizations. This is similar to the function provided by the MQ V6 and MQ 7.0 SupportPac MS03 “saveqmgr” (which does not work with MQ 7.1 and later).

However, this dmpmqcfg command does NOT export the AMS Policies.
Thus, in case that you want to export these AMS Policies into a file that is part of backup procedures, you can use the command “dspmqspl” with the -export flag, as shown below.

Export the policies in the form of of a setmqspl command that could be executed later, in case that you need to restore the policies:

dspmqspl -m QM_VERIFY_AMS -export > restore_AMS_policies.bat

Display the output file:

cat restore_AMS_policies.bat
setmqspl -m QM_VERIFY_AMS -p TEST.Q -s SHA1 -a "CN=alice,O=IBM,C=GB" -e AES25r "CN=bob,O=IBM,C=GB" -t 0

++
+++ Chapter 8: Basic testing of the setup
++

Let's test the setup by putting a message as user alice and reading the message as user bob.

+ Window 2: User alice

As user alice, put a message using a sample application. Type the text of the message, then press Enter.

amqsput TEST.Q QM_VERIFY_AMS
Sample AMQSPUT0 start
target queue is TEST.Q
this is a test

Sample AMQSPUT0 end

+ Window 3: User bob

As user bob, get a message using a sample application:

amqsget TEST.Q QM_VERIFY_AMS
Sample AMQSGET0 start
message <this is a test>
no more messages
Sample AMQSGET0 end

Conclusion: User alice was able to put a message and bob was able to read it.

++
+++ Chapter 9: Testing encryption
++

To verify that the encryption is occurring as expected, create an alias queue which references the original queue TEST.Q.

This alias queue will have no security policy and so no user will have the information to decrypt the message and therefore the encrypted data will be shown.

+ Window 1: User mqm

Create an alias queue

runmqsc QM_VERIFY_AMS
   DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
   end

Grant bob access to browse from the alias queue

setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse

+ Window 2: User alice

As user alice, put another message:

amqsput TEST.Q QM_VERIFY_AMS

+ Window 3: User bob

As user bob, browse the message via the alias queue:

amqsbcg TEST.ALIAS QM_VERIFY_AMS

The output from amqsbcg application shows the encrypted data that is on the queue proving that the message has been encrypted:

+ begin output
AMQSBCG0 - starts here
****

```
 MQOPEN - 'TEST.ALIAS'


 MQGET of message number 1, CompCode:0 Reason:0
****Message descriptor****

  StrucId  : 'MD  '  Version : 2
  Report   : 0  MsgType : 8
  Expiry   : -1  Feedback : 0
  Encoding : 546  CodedCharSetId : 1208
  Format : '        '
  Priority : 0  Persistence : 0
  MsgId : X'414D5120514D5F5645524946595F414D3993075305310020'
  CorrelId : X'00000000000000000000000000000000000000000000000000000000'
  BackoutCount : 0
  ReplyToQ       : '                                                '
  ReplyToQMgr    : 'QM_VERIFY_AMS                                 '
  ** Identity Context
  UserIdentifier : 'alice       '
  AccountingToken :
   X'0000000000000000000000000000000000000000000000000000000000000000'
  ApplIdentityData : '                                '
  ** Origin Context
  PutApplType    : '6'
  PutApplName    : 'amqsput                   '
  PutDate  : '20140224'    PutTime  : '15040661'
  ApplOriginData : '    '

  GroupId : X'000000000000000000000000000000000000000000000000'
  MsgSeqNumber   : '1'
  Offset         : '0'
  MsgFlags       : '0'
  OriginalLength : '-1'

****   Message      ****

 length - 1238 of 1238 bytes

00000000:  5044 4D51 0200 0200 6800 0000 6800 0000          'PDMQ....h...h...'
00000010:  0800 0000 B804 0000 0E00 0000 0000 0000          '................'
00000020:  4D51 5354 5220 2020 0000 0000 0000 0000          'MQSTR   ........'
00000030:  0000 0000 0000 0000 2020 2020 2020 2020          '........        '
00000040:  2020 2020 2020 2020 2020 2020 2020 2020          '                '
```

```
00000050:  2020 2020 2020 2020 2020 2020 2020 2020          '               '
00000060:  2020 2020 2020 2020 3082 046A 0609 2A86          '       0..j..*.'
00000070:  4886 F70D 0107 03A0 8204 5B30 8204 5702          'H.........[0..W.'
...
000004A0:  D99F 93C9 1E16 535D 2B30 C141 495B 7548          'Ù....S]+0.AI[uH'
000004B0:  7264 7CFB A471 D6DD 6576 36BC FA99 5D95          'rd|..q..ev6...].'
000004C0:  0148 B66D 4FD4 B581 B1AB 63CC 11F8 5A0F          '.H.mOÔµ...c...Z.'
000004D0:  F0C2 6B2C E73E                                   '..k,.>         '
```

No more messages
MQCLOSE

+ end output

++
+++ Chapter 10: Advanced testing
++

+++ Scenario A: not authorized by AMS to view messages

Let's explore what happens when other users, who are not authorized explicitly to use the queues protected by AMS, try to view the messages.

+ Window 2: User alice

As user alice, put a message using a sample application. Type the text of the message, then press Enter.

amqsput TEST.Q QM_VERIFY_AMS
Sample AMQSPUT0 start
target queue is TEST.Q
this is a test

Sample AMQSPUT0 end

+ Window 1: User fulano

Log in as user fulano and ensure to set up the environment for using MQ 7.5:
. set-mq-75.ksh

Try to put, browse or get a message from the queue. These actions will fail.

Even though the setmqaut was given for user fulano to get messages from the queue TEST.Q, the AMS policies do not include user fulano as an authorized user:

amqsput TEST.Q QM_VERIFY_AMS
Sample AMQSPUT0 start
target queue is TEST.Q
MQOPEN ended with reason code 2035
unable to open queue for output
Sample AMQSPUT0 end

amqsbcg TEST.Q QM_VERIFY_AMS
AMQSBCG0 - starts here
****
MQOPEN - 'TEST.Q'

MQOPEN failed with CompCode:2, Reason:2035

amqsget TEST.Q QM_VERIFY_AMS
Sample AMQSGET0 start
MQOPEN ended with reason code 2035
unable to open queue for input
Sample AMQSGET0 end

Notice that the reason code is 2035. You can use the following MQ command to get the short name for a reason code, in order to get a rough idea of that the problem is:

mqrc 2035
2035 0x000007f3 MQRC_NOT_AUTHORIZED

+ Window 1: User mqm

Log in as user mqm

As user mqm try to browse the message:

amqsbcg TEST.Q QM_VERIFY_AMS

AMQSBCG0 - starts here
****
MQOPEN - 'TEST.Q'
MQOPEN failed with CompCode:2, Reason:2035

NOTE:
The user mqm, even though it is an MQ administrator, is NOT authorized to read the messages.

+ Error messages in the queue manager error log

Let's look at the error messages in the queue manager error log:

cd /var/mqm/qmgrs/QM_VERIFY_AMS/errors

tail AMQERR01.LOG

+ begin excerpt
02/24/2014 07:42:50 AM - Process(6877.1) User(fulano) Program(amqsput)

Host(veracruz) Installation(Installation2)
VRMF(7.5.0.3) QMgr(QM_VERIFY_AMS)

AMQ9062: The WebSphere MQ security policy interceptor could not read the keystore configuration file: /home/fulano/.mqs/keystore.conf.

EXPLANATION:
The WebSphere MQ security policy interceptor could not read the keystore configuration file: /home/fulano/.mqs/keystore.conf.
ACTION:
Make sure that the user who executes the WebSphere MQ application has permissions to read the configuration file. Check if the configuration file is not corrupted or empty. If the problem persists, contact your local IBM service representative.

+ end excerpt

+++ Trying to view the queue files directly

In the following scenario, a new normal queue Q1 will be created, and a message will be put there. Then the actual queue files handled by the queue manager will be viewed.

+ Window 1: user mqm

Create a local queue Q1:

runmqsc QM_VERIFY_AMS
define qlocal(Q1)
end

Put a message into Q1:

amqsput Q1 QM_VERIFY_AMS
Sample AMQSPUT0 start
target queue is Q1
this is a test

Sample AMQSPUT0 end

List the file sizes for the queues TEST.Q and Q1:

cd /var/mqm/qmgrs/QM_VERIFY_AMS/queues

ls -l TEST* Q1
-rw-rw---- 1 mqm mqm 2560 2014-02-24 07:56 Q1
-rw-rw---- 1 mqm mqm 4096 2014-02-24 07:56 TEST!Q

Notice that the queue TEST.Q has a file that is called TEST!Q.
It might seem strange to Unix and Windows users to replace the . (dot, period) with an exclamation point (!) in the file name, however it is necessary due to the multi-platform support in MQ (iSeries, z/OS)

Notice also that even though both queues have only 1 message, with the text “this is a test”, the file size for TEST!Q (protected by AMS) is larger than for Q1 (normal queue).

Let display the contents of Q1. Notice that we can figure out the text (shown in bold and underlined): this is a test

cat Q1

+ begin output
AQQF° ¸$Äÿÿÿ
ÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿLLQQ1
2014-02-24 07.30.42
@   ÿÉ;ÿÿÿÿ2014-02-24 07.30.42
8àcÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿAQRHÿÿÿÿÿÿÿÿÿÿÿÿÿÀAMQ QM_VERIFY_AM9S.
ÿÿÿÿ&óY98}ÿÿÿÿMD "¸MQSTR   QM_VERIFY_AMS
rivera   amqsput   2014022412554198   ÿÿ**<u>this is a test</u>**¸MQSTR
+ end output

Let's do the same for the file TEST!Q. Actually you need to enclose the file name between single quotes.
Notice that we cannot see the plain text “this is a test” because it is encrypted!

cat 'TEST.Q'

+ begin output
AQQF° ¸$Äÿÿÿ
ÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿLLQTEST.Q
2014-02-21 12.56.21

@ ÿÉ;ÿÿÿÿ2014-02-21 12.56.21
PÿÉ;ýÿÿÿÿýÿÿÿÿýÿÿÿÿýÿÿÿÿ
ÿÿÿÿDßEUÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿAQRHÿÿÿÿÿÿÿÿÿÿÿÿÿÀÖAMQ QM_VERIFY_AM9S,
ÿÿÿÿ&ó°3qÿÿÿÿMD "¸ QM_VERIFY_AMS
alice amqsput 2014022412522036
ÿÿPDMQh¸MQS [0W1Ï0Ì050)1 0j *H÷
0 UGB1
0
U
IBM1
0 ;Ò\+¤´c9ë
ã>¬õ´]ÂÑ\Ï0
S Ï,é(7¹ÇNÚÎYzÿÀqVñTì#×GÁGhsÙ´tQ3êZe²Ø
~¾ëµôÎ
DÏnó>*~Ó]´4@{Z)%p q$h*ÄFgLÝç¤Møµ±ÏQãC+¦~Ïsdî2:FkÎr{Ur
jº½:Ht@VÄÍssQCHíøÜ:ÝG@Ä9íÿ¥@þQæ!ÛþóôGNP9ÓÐOM,Ëntt²kS}íêfmkal?
_:8GÊ1GÒ¯îöÏi¥Bºýbê©2§HªJ%I
6Ã]È¡Ë$-omþ=QYU]¡
ÐpAS³¶oª:ú:YVTUk±¼ÐtS°Áñ3~»¦±zV#èÒ²?¤H{:Éjí°4Ï_ÁS¿
¬ü~ïº#ìÛçò®C8
ÐÈçI@@d°Í/AÍ[T ¿þlÉMÆ ®§Mç
-¼Ó^Ià'ý´Å&3·^/þíÄ¯Sâ2ðÉ»Il*.«soó=¿Cò}nY`®TîÍíâsC¯±éZ4X¨ãÞzÏ>-
à#$ª{õÜ'«çµ1Ì¶áØúÕIucMúÎøïù3·Y!ÕÇØ7Ä½
;ãõ*HTxÈ¦N/Bl'¯Û¤þx½uÜ|}©
ø¨æ3¨sæHñp;?
nAfiVDS£âJ$b¦ßºÒC}ú"û×ái +
£FH.Á)£·¯ÎxõÒÖºÝÓÌamqzfuma 2014022117564227
ÿÿ>zfdLISTENER.1444

+ end output

Conclusions:
- Only users alice and bob, who are fully authorized to put/get messages in the TEST.Q are allowed to put and get messages.
- Not even the user “mqm”, who is MQ administrator is able to browse, put or get messages from the protected queue TEST.Q
- The messages are encrypted in the file used by the queue manager for the protected queue, and even with direct access to the physical file, it is not possible view the payload/contents of the protected message.

+++ Scenario B: User alice is not authorized by AMS to read messages signed by bob

Only one AMS policy has been created for this technical document.
In this policy the user "alice" was explicitly indicated as a "signer" and user "bob" was indicated as a "reader".

Now, let's explore the following scenario, which is NOT covered by the above policy: the user "bob" puts a message as a signer and user "alice" tries to read it.
Because there is no explicit policy for this case, the error message that we get will be 2063:

2063 0x0000080f MQRC_SECURITY_ERROR

Window 3 (bob)

As user bob put a message into TEST.Q. This is successful.
The message is encrypted and placed encrypted in the queue.

$ **amqsput TEST.Q QM_VERIFY_AMS**
Sample AMQSPUT0 start
target queue is TEST.Q
t2

Sample AMQSPUT0 end

Window 2 (alice)

As user alice try to browse message from TEST.Q.
This is not successful.

$ **amqsbcg TEST.Q QM_VERIFY_AMS**
AMQSBCG0 - starts here
****
MQOPEN - 'TEST.Q'
MQGET 1, failed with CompCode:2 Reason:2063
MQCLOSE

The reason code 2063 means: MQRC_SECURITY_ERROR

It is necessary to view the queue manager error log to get more details.
The last item, number 4, is the one that applies to this situation:

(4) receiver is not among the recipients of the message.

05/28/2014 12:34:20 PM - Process(6256.1) User(alice) Program(amqsbcg)
Host(veracruz) Installation(Installation2)
VRMF(7.5.0.3) QMgr(QM_VERIFY_AMS)

AMQ9017: WebSphere MQ security policy internal error: message could not be unprotected: GSKit error code 851968, reason 43.

EXPLANATION:
The WebSphere MQ security policy interceptor could not verify or decrypt a message because the indicated GSKit error occurred. This can happen for several reasons, all of which are internal failures:
(1) the message is not a valid PKCS#7 message;
(2) the sender's certificate does not have the required key
usage bit to be able to encrypt the message;
(3) the sender's certificate was not recognized as a trusted certificate;
(4) receiver is not among the recipients of the message.
ACTION:
Consult the GSKit information in the Information Center for the explanation of the GSKit reason code and take corrective action. If the problem persists, contact your IBM service representative.

+++ Scenario C: User bob is not authorized by AMS to read messages signed by bob

As mentioned in the previous scenario in this chapter, only one AMS policy has been created for this technical document.
In this policy the user "alice" was explicitly indicated as a "signer" and user "bob" was indicated as a "reader".

Now, let's explore the following scenario, which is NOT covered by the above policy: the user "bob" puts a message as a signer and the same user "bob" tries to read it. Because there is no explicit policy for this case, the error message that we get will be 2063:

```
   2063  0x0000080f  MQRC_SECURITY_ERROR
```

This may seem a bit strange: unless there is a policy in place, user bob CANNOT browse the encrypted messages generated by himself!

Window 3 (bob)

As user bob put a message into TEST.Q. This is successful.
The message is encrypted and placed encrypted in the queue.

```
$ amqsput TEST.Q QM_VERIFY_AMS
Sample AMQSPUT0 start
target queue is TEST.Q
t2
Sample AMQSPUT0 end
```

Now, again as user bob, try to browse the message:

```
$ amqsbcg  TEST.Q QM_VERIFY_AMS
AMQSBCG0 - starts here
**********************
 MQOPEN - 'TEST.Q'
 MQGET 1, failed with CompCode:2 Reason:2063
```

Let's take a look at the queue manager error log to get more details:

```
05/28/2014 12:33:53 PM - Process(5760.1) User(bob) Program(amqsbcg)
                  Host(veracruz) Installation(Installation2)
                  VRMF(7.5.0.3) QMgr(QM_VERIFY_AMS)
```

AMQ9035: Message signer is not in the list of authorised signers.

EXPLANATION:
The WebSphere MQ security policy interceptor detected that the message is signed by an unauthorised party.
ACTION:
Establish whether the identity associated with the sender of the message is authorized to send messages to this application. Ensure the sender is named in the list of allowed signers on the security policy for the queue.

+++ end +++