How to configure in MQ the mqccred client side security exit

https://www.ibm.com/support/pages/node/6538020

Date last updated: 03-Jan-2022

Miguel Rodriguez
IBM MQ Support
https://www.ibm.com/products/mq/support
Find all the support you need for IBM MQ

Acknowledgement:
Thanks to Mike Cregger for compiling the original material in this document.

Question:

How to pass a userId and a password using the mqccred security exit provided with the MQ Version 8 and later products.

Cause:

You have a legacy application and you are required to pass a userId and password for authentication with the queue manager.

Answer:

MQ provides a security exit called **mqccred** that can be used to pass a userId and password. The mqccred security exit is deployed on the client side, therefore, in order to make this work, the mqccred exit programs and Client Channel Definition Table (CCDT) file will need to be copied to the client side system.

The mqccred security exit will need to be referenced in the corresponding client connection definition in the CCDT file.

Instructions:

1. Ensure you have the correct mqccred security exit executables/libraries for your Operating System platform and the correct 32-bit/64-bit library per your MQ client application.
These libraries are provided with the MQ client samples, under tools/c/mqccred sub-directory.
They should be copied to the MQ client's exits/exits64 sub-directory which is contained on the MQ client's or MQ Server data sub-directory.

For example, in MQ 9.2 in Windows, the directory is:
C:\Program Files\IBM\MQ\tools\c\Samples\mqccred

For example, in MQ 9.2 in Linux, the directory is:
/opt/mqm/samp/mqccred

2. Create a file **mqccred.ini** with your userid/password.

3. Use the program **runmqccred** to obfuscate the password(s) in the file.

4. Modify the CLNTCONN channel to reference the security exit. This needs to be configured before the client-side channel definition table (CCDT) file is copied to the MQ Client system.

5. Ensure that the file permissions on the mqccred.ini file are such that ONLY certain user(s) and group(s) have access to the file. **The file CAN NOT be readable by all.**

WARNING: On the Windows platform, ensure the mqccred.ini file DOES NOT have inherited permissions and only specific users and groups have access to the file.

6. Copy the mqccred and mqccred_r exit programs, which are provided as samples with MQ to the /var/mqm/exits(32 bit) or /var/mqm/exits64(64 bit) subdirectory on the client system.

For example:
If the MQ client application is running on a Windows platform, ensure to get a Windows' version of the mqccred.dll from the Windows MQ client or MQ Server install, and ensure it is copied to the MQ client's exits (for 32 bit applications) or exits64 (for 64 bit applications) subdirectory.

7. If using a CCDT, ensure your client's CCDT tab file is updated to include the correct client channel definition.

8. Ensure your client application knows where to find the CCDT file, by setting (Windows) or exporting (Linux/Unix) 2 environment variables according to the article below.
MQCHLLIB and MQCHLTAB

https://www.ibm.com/docs/en/ibm-mq/9.2?topic=tables-configuring-binary-format-ccdt
IBM MQ / 9.2 /
Configuring a binary format CCDT

9. The location of the mqccred.ini file (MQCCRED) can also be set or exported.

For example on Windows:
```
set MQCCRED=C:\YourDirectory\mqccred.ini
```

Configuration and Usage:

Below are detailed instructions configuring and using the mqccred security exit on Linux and Windows Systems.

1. Linux - Using the MQ amqsputc and amqsgetc sample applications to test the mqccred exit:

Note: In this example the MQ Client and Queue Manager are running on the same system.

a. Create test QM:

crtmqm QM80A

strmqm QM80A

b. Temporarily disable CHLAUTH

runmqsc QM80A

alter qmgr CHLAUTH(DISABLED)

c. Create Listener -

DEFINE LISTENER(LIST1) TRPTYPE(TCP) PORT(1802) CONTROL(QMGR)
START LISTENER(LIST1)

d. Create channels and a queue –

DEFINE CHANNEL(CHAN1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)

DEFINE CHANNEL(CHAN1) CHLTYPE(CLNTCONN) CONNAME ('a.b.c.d(1802)') QMNAME(QM80A) SCYEXIT('mqccred(ChlExit)') SCYDATA(DEBUG)

DEFINE QLOCAL(Q1)

e. Copy mqccred or mqccred_r programs to /var/mqm/exits64 directories.

NOTE:
Make sure to copy the correct ones.. lib64(64-bit) or lib(32-bit)
For Example on my RHEL 64 bit OS..

cp /opt/mqm80/samp/mqccred/lib64/* /var/mqm/exits64

f. Here is an example of the mqqcred.ini file **before** the passwords were obfuscated:

```
AllQueueManagers:
User=testuser
Password=passw0rd

QueueManager:
Name=QM80A
User=mqm
Password=passw0rd
```

g. To run amqsputc/getc as mqm user with mqccred

Login as mqm

Set installation to MQv8

$. /opt/mqm80/bin/setmqenv -s

Ensure AMQCLCHL.TAB with client channel and mqqcred exit defined is in the local subdirectory.

Ensure mqccred.ini is in the local subdirectory

Make a directory - mkdir mqccred-test

Change directory - cd mqccred-test

Copy the CCDT file - cp /var/mqm/qmgrs/QM80A/@ipcc/AMQCLCHL.TAB .

h. At this point, the mqccred.ini and AMQCLCHL.TAB files should be in the current directory, export the environment variables.

```
$ unset MQSERVER
$ export MQCHLLIB=./
$ export MQCHLTAB=AMQCLCHL.TAB
$ export MQCCRED=./mqccred.ini
```

i. Run the amqsputc sample application –

```
$ /opt/mqm80/samp/bin/amqsputc Q1 QM80A
Sample AMQSPUT0 start
mqccred exit:
Configuration file is at ./mqccred.ini.
Accessible: Yes
mqccred exit: Searching for queue manager 'QM80A'
mqccred exit: Configuration file at ./mqccred.ini contains a plaintext password
mqccred exit: ReadConfigFile rc = 1004 MQCONNX ended with reason code
2537
```

NOTE: The full error reported is due to the SCYDATA(DEBUG) in the client connection

channel definition. If the DEBUG value was not defined on the channel, the connection failure would only report the 2537 error.

You must run the runmqccred command to obfuscate the passwords in the mqccred.ini file, for example:

```
$ /opt/mqm80/samp/mqccred/runmqccred
File './mqccred.ini' processed successfully.
Plaintext passwords found: 2
```

j. Run the amqsputc sample application again–

```
$ /opt/mqm80/samp/bin/amqsputc Q1 QM80A

Sample AMQSPUT0 start
mqccred exit: Configuration file is at ./mqccred.ini.
Accessible: Yes
mqccred exit: Searching for queue manager 'QM80A'
mqccred exit: Returning info for user 'mqm', forceOverride=1
mqccred exit: ReadConfigFile rc = 0 target queue is Q1 test123
Sample AMQSPUT0 end
```

2. Windows - Using the MQ amqsputc and amqsgetc sample applications to test the mqccred exit:

a. Create test QM:

crtmqm QM80A

strmqm QM80A

b. Temporarily disable CHLAUTH -

runmqsc QM80A

alter qmgr CHLAUTH(DISABLED)

c. Create a Listener-

DEFINE LISTENER(LIST1) TRPTYPE(TCP) PORT(1802) CONTROL(QMGR)

START LISTENER(LIST1)

d. Create channels and a queue –

DEFINE CHANNEL(CHAN1) CHLTYPE(SVRCONN)

DEFINE CHANNEL(CHAN1) CHLTYPE(CLNTCONN) CONNAME ('a.b.c.d(1802)') QMNAME(QM80A) SCYEXIT('mqccred(ChlExit)') SCYDATA(DEBUG)

DEFINE QLOCAL(Q1)

e. Copy the mqccred.dll (32 and 64bit versions)

NOTE:
Make sure to copy the correct ones.. lib64(64-bit) or lib(32-bit)

For example on my Windows OS.. my qmgrs data directory is on C:\IBM\WMQ

copy [MQInstalldir]\Tools\c\Samples\mqccred\bin64\mqccred.dll C:\ibm\wmq\exits64\Installation1

f. Here is example of my mqccred.ini file:

```
AllQueueManagers:
User=testuser
Password=passw0rd

QueueManager:
Name=QM80A
User=JoeTest
Password=passw0rd
```

g. Disable inheritance.

Right click on the mqccred.ini file and select properties.

Click on the “Security” tab.

Click on the “Advanced” button.

Click on the “Disable inheritance” button.

Remove Authenticated and User groups from the list.

Leave the Adminstrators and SYSTEM.

Add whichever User you would like and select authorizations.

Press OK.

h. To run amqsputc/getc as a member of the mqm local group or administrator's group with mqccred:

Login as a userId in the mqm local group or administrators group,

Set installation - set installation to MQv8 setmqenv -s

Ensure AMQCLCHL.TAB with client channel with mqqcred defined is in the local subdirectory

Ensure mqccred.ini is in the local subdirectory

Make a directory - mkdir c:\test\mqccred

Change directory - cd c:\test\mqccred

Copy the CCDT –

copy C:\ibm\wmq\qmgrs\QM80A\@ipcc\AMQCLCHL.TAB .

i. At this point mqccred.ini and AMQCLCHL.TAB should be in the current directory set the following environment variables.

```
set MQSERVER=
set MQCHLLIB=.\
set MQCHLTAB=AMQCLCHL.TAB
set MQCCRED=.\mqccred.ini
```

j. Run the amqsputc sample application again–

```
C:\test\mqccred>amqsputc Q1 QM80A
Sample AMQSPUT0 start
mqccred exit: Configuration file is at ./mqccred.ini.
Accessible: Yes
mqccred exit: Searching for queue manager 'QM80A'
mqccred exit: Returning info for user 'mqm', forceOverride=1
mqccred exit: ReadConfigFile rc = 0 target queue is Q1 test123
Sample AMQSPUT0 end
```

k. Example of an Error:

```
C:\test\mqccred>amqsputc Q1 QM80A
Sample AMQSPUT0
mqccred exit: Configuration file is at ./mqccred.ini.
Accessible: Yes
mqccred exit: Searching for queue manager 'QM80A'
mqccred exit: Configuration file at ./mqccred.ini contains a plaintext password
mqccred exit: ReadConfigFile rc = 1004 MQCONNX ended with reason code
2537
```

NOTE: The full error reported is due to the SCYDATA(DEBUG) in the client connection channel definition.

If the DEBUG value was not defined on the channel, the connection failure would only report the 2537 error.

As the error indicates, the file permissions are not set correctly. Ensure the OS file permissions on the mqccred.ini are NOT inherited(step g), and only certain user(s) and/or group(s) have access.

You must run the runmqccred to obfuscate the passwords in the mqccred.ini file:

```
C:\test\mqccred>\ibm\mq9\tools\c\samples\mqccred\runmqccred File './mqccred.ini'
processed successfully.
 Plaintext passwords found: “Number”
```

+++ end