IBM MQ Security: Overview & recap

Rob Parker, IBM

parrobe@uk.ibm.com

Important Disclaimer

- **THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.**
- **WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.**
- **IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.**
- **IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.**
- **NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:**
  - CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
  - ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.
- **Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.**

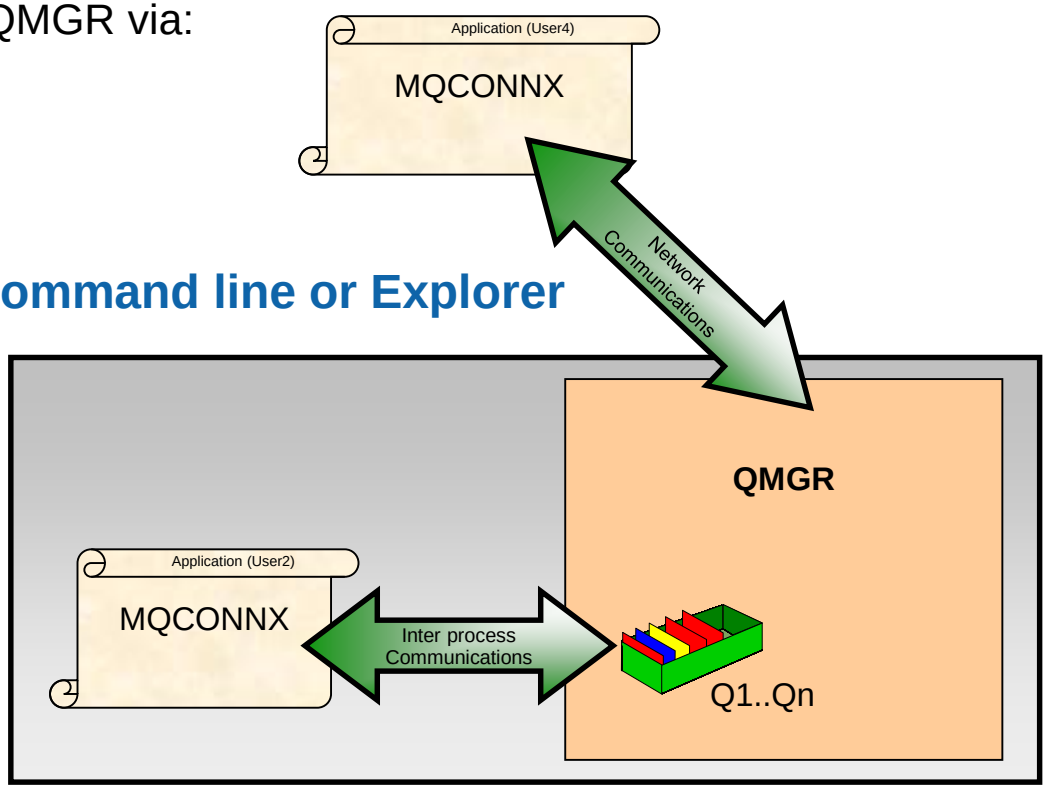
Agenda

- Introduction
- Connection Authentication
- Authorization
- Transport Layer Security
- Channel Authentication
- Security Exits
- AMS

Introduction – Typical MQ

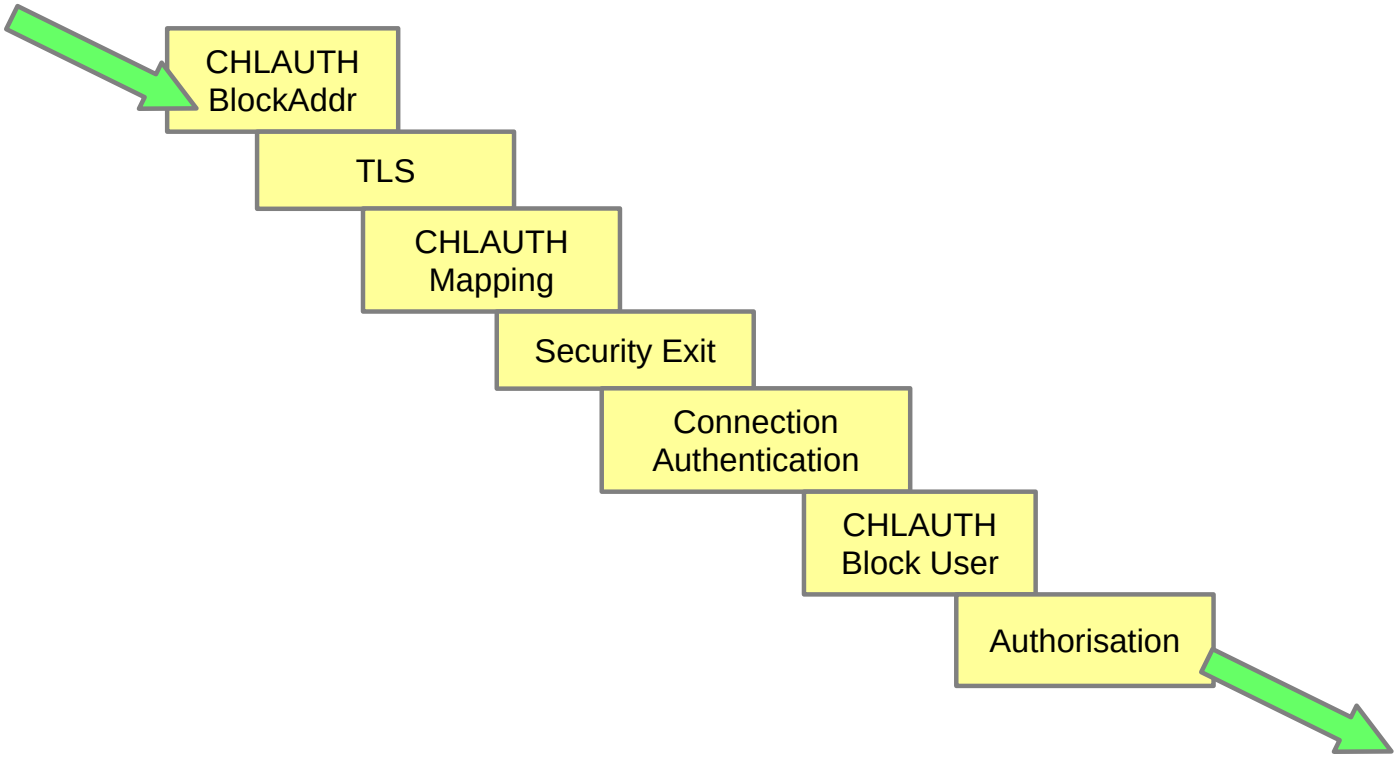
- **In a Typical MQ setup there is:**
  - A Queue Manager (QMGR)
  - A number of Queues
  - Applications that connect to the QMGR via:
    - Local Bindings
    - Client connections

- **Configuration is updated via Command line or Explorer**



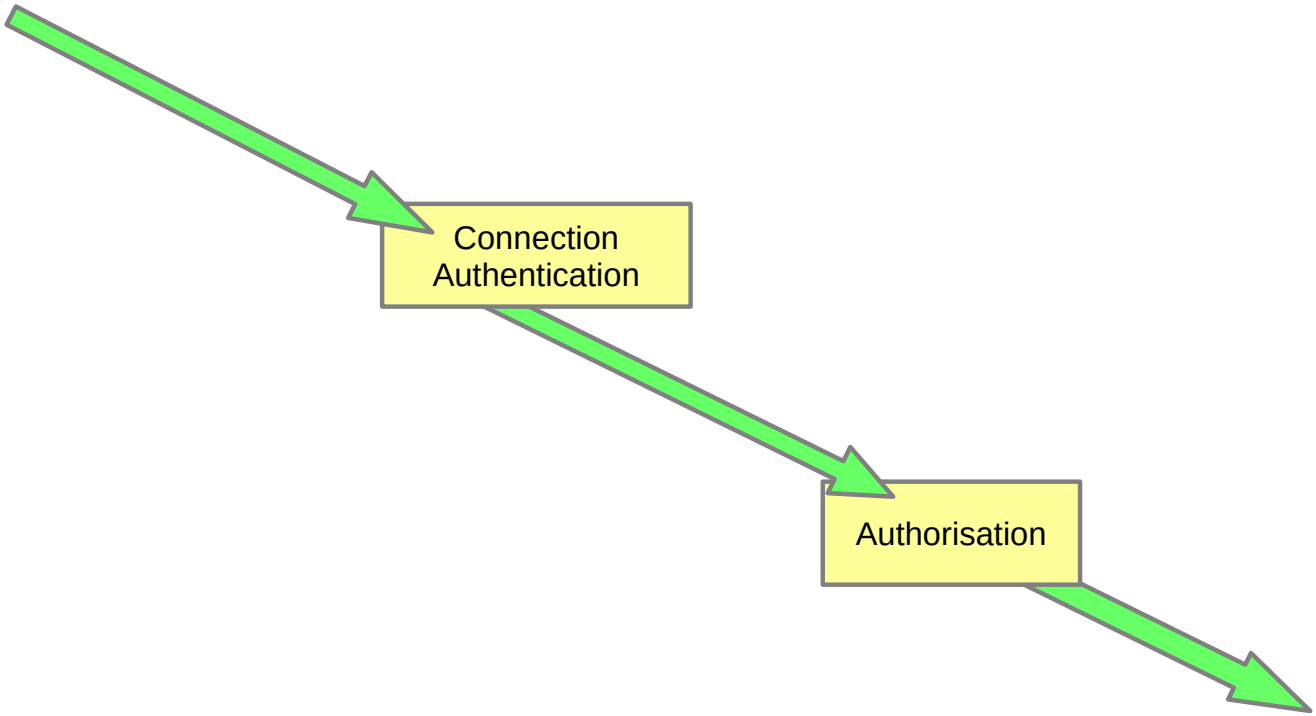
Introduction – Security Checks (Client)

- **When a user connects via client:**



Introduction - Security Checks (Local)

- **When a user connects via local bindings:**



AUTHENTICATION

Details

- **We use Authentication to ask clients connecting to prove they are who they say they are.**
  - Usually used in combination with authorisation to limit user's abilities.

- **Connection authentication feature available in MQ v8 and above.**
  - Allows authentication using user credentials supplied by client applications.
  - User credentials can be local OS users or LDAP users.
  - A failure to authenticate results in a MQRC_NOT_AUTHORIZED 2035 error being returned.

- **IBM MQ now can send two different userids in the connection data.**
  - The userid that is running the application.
  - The userid and password that the application wants to authenticate with.

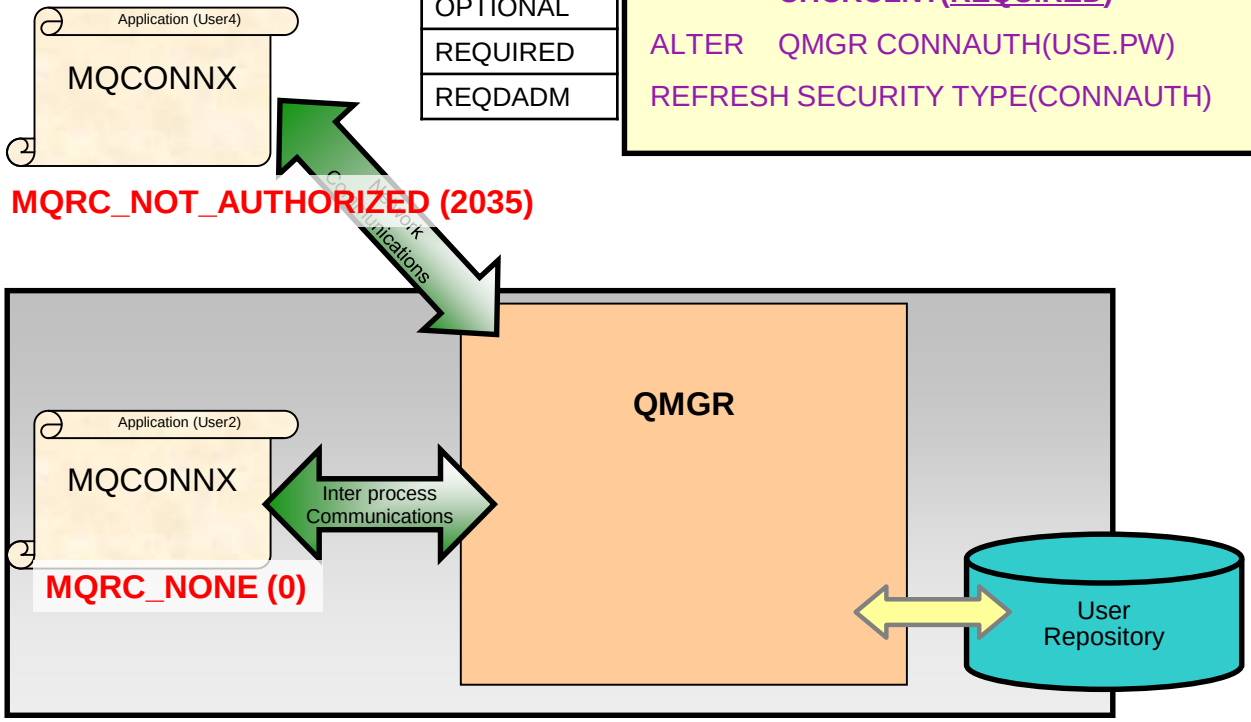
Configuration

CHCK…
NONE
OPTIONAL
REQUIRED
REQDADM

DEFINE AUTHINFO(USE.PW) AUTHTYPE(*xxxxxx*)
CHCKLOCL(OPTIONAL)
CHCKCLNT(REQUIRED)

ALTER QMGR CONNAUTH(USE.PW)

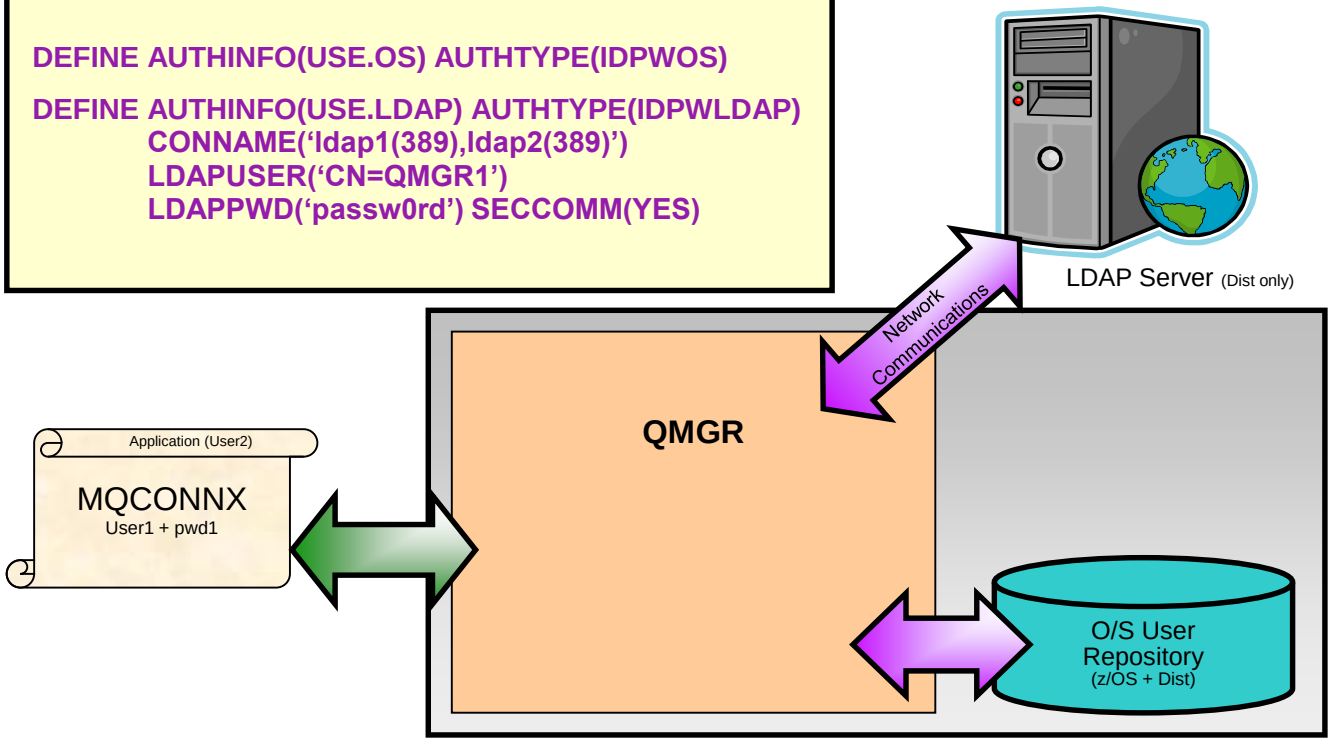
REFRESH SECURITY TYPE(CONNAUTH)



Configuration

```
DEFINE AUTHINFO(USE.OS) AUTHTYPE(IDPWOS)

DEFINE AUTHINFO(USE.LDAP) AUTHTYPE(IDPWLDAP)
        CONNAME('ldap1(389),ldap2(389)')
        LDAPUSER('CN=QMGR1')
        LDAPPWD('passw0rd') SECCOMM(YES)
```



AUTHORIZATION

Details

- **We use Authorization to limit what connected users can and cannot do.**

- **This is performed by creating authority records**
  - We create authority records for a specific user or group.
  - User level authority records are available on Linux but not by default

- **Authority is given on MQ objects and dictate what actions they can performed (PUT, GET, OPEN, etc)**

- **If a user or group does not have authority to do what they are trying to do, they get blocked.**
  - MQRC_NOT_AUTHORIZED (2035)
  - Users who are members of the mqm group have full administrator access.

- **A channel or channel authentication rule can change the userid used for authority checks**

Which user will be used for authorization?

Method	Notes
Client machine user ID flowed to server	This will be over-ridden by anything else. Rarely do you want to trust an unauthenticated client side user ID.
MCAUSER set on SVRCONN channel definition	A handy trick to ensure that the client flowed ID is never used is to define the MCAUSER as ‘rubbish’ and then anything that is not set appropriately by one of the next methods cannot connect.
MCAUSER set by CHLAUTH rule	To allow more granular control of MCAUSER setting, rather than relying on the above queue manager wide setting, you can of course use CHLAUTH rules
MCAUSER set by ADOPTCTX(YES)	The queue manager wide setting to adopt the password authenticated user ID as the MCAUSER will over-ride either of the above.
MCAUSER set by Security Exit	Although CHLAUTH gets the final say on whether a connection is blocked (security exit not called in that case), the security exit does get called with the MCAUSER CHLAUTH has decided upon, and can change it.

Again with Early Adopt

Method	Notes
Client machine user ID flowed to server	This will be over-ridden by anything else. Rarely do you want to trust an unauthenticated client side user ID.
MCAUSER set on SVRCONN channel definition	A handy trick to ensure that the client flowed ID is never used is to define the MCAUSER as 'rubbish' and then anything that is not set appropriately by one of the next methods cannot connect.
MCAUSER set by ADOPTCTX(YES)	The queue manager wide setting to adopt the password authenticated user ID as the MCAUSER will over-ride either of the above.
MCAUSER set by CHLAUTH rule	To allow more granular control of MCAUSER setting, rather than relying on the above queue manager wide setting, you can of course use CHLAUTH rules
MCAUSER set by Security Exit	Although CHLAUTH gets the final say on whether a connection is blocked (security exit not called in that case), the security exit does get called with the MCAUSER CHLAUTH has decided upon, and can change it.

Configuration

- **Authority records are created or modified using one of three tools:**

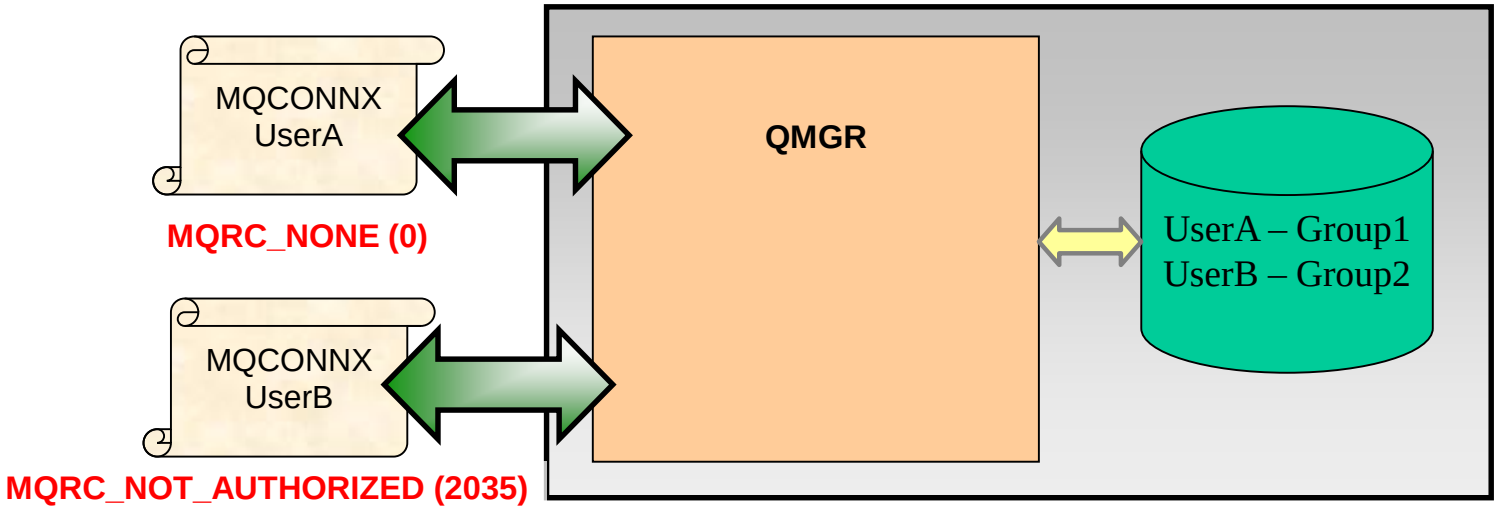
- **runmqsc**
  - `SET AUTHREC(<Object name>) OBJTYPE(<Object type>) GROUP(<group name>) AUTHADD|AUTHRMV(<authority to give|remove>)`

- **setmqaut**
  - `setmqaut –m <QM name> -n <Object name> -t <Object type> -g <Group name> <authorizations to give or remove>`

- **IBM MQ Explorer**
  - By right clicking on the object you want to grant/remove authorities for and selecting "`Object Authorities -> Manage Authority Records`"

Configuration

```
SET AUTHREC  OBJTYPE(QMGR)        +
             GROUP('Group1')      +
             AUTHADD(CONNECT)
```



TRANSPORT LAYER SECURITY (TLS)

Details

- **TLS uses Private-Public asymmetric keys to exchange symmetric keys used to encrypt data.**
  - The symmetric keys exchanged are referred to as “session keys”.
  - The asymmetric keys are associated with a certificate that is used for identity.

- **IBM MQ’s integration of TLS provides the following two features:**
  - Encryption of transmissions between client/queue manager to queue manager.
  - [optional] Authentication with a queue manager.

Details

- **Certificates are created, stored and managed using tools supplied with IBM MQ**
  - runmqakm
  - runmqckm
  - iKeyman (strmqikm)

- **Certificates must be stored in a keystore format recognised by the queue manager (CMS)**
  - The keystore password must also be available to the queue manager in a secure stash file.

- **IBM MQ Channels can only have a single CipherSpec set on them**
  - A CipherSpec is a string which details the hashing and encryption algorithm to use.
  - A list of the cipher strings you can supply are detailed on the knowledge centre.

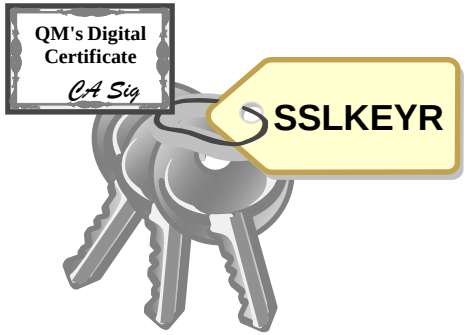
Details

- **IBM MQ allows clients to either connect anonymously or with mutual authentication**
  - If a client connects with a certificate then it must be known and trusted by the queue manager.

- **CipherSpec lists are updated when new vulnerabilities arise**
  - In later versions of IBM MQ you may notice the list size changing.
  - We do not delete CipherSpecs, we disable them by default.

- **MQv8 added in multiple certificates feature**
  - Allows you to specify a different certificate to use at the channel level
  - Allows you to specify a certificate to use on the queue manager
    - Before you would be forced to name your certificate **`ibmwebspheremq<QM name>`**

Configuration



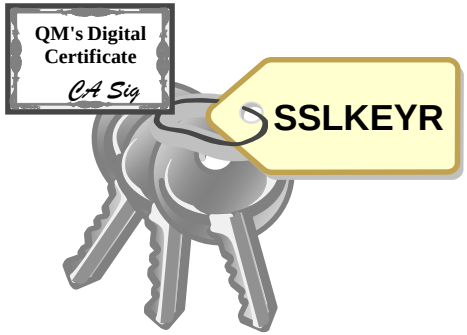
- **Once you have created a Key store for the server to use:**
  - `ALTER QMGR SSLKEYR(<location of keystore>)`

- **Once you have created the certificate for the server to use (MQv8+ only)**
  - `ALTER QMGR CERTLABL(<certificate label>)`

ALTER QMGR SSLKEYR('var/mqm/qmgrs/QM1/ssl/key') CERTLABL('QM1Certificate')

REFRESH SECURITY TYPE(SSL)

Configuration



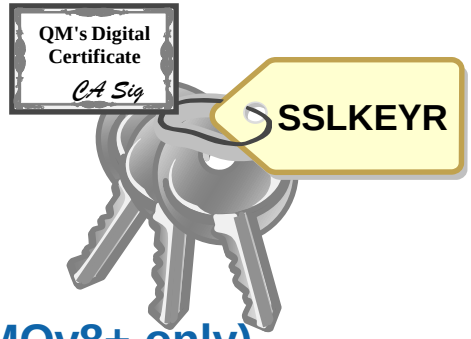
- **Once you have created a Key store for the server to use:**
  - `ALTER QMGR SSLKEYR(<location of keystore>)`

- **Once you have created the certificate for the server to use (MQv8+ only)**
  - `ALTER QMGR CERTLABL(<certificate label>)`

- **To enable TLS on a channel, specify a CipherSpec to use.**
  - `ALTER CHANNEL(<channel name>) CHLTYPE(<channel type>) SSLCIPH(<Cipher string>)`

ALTER CHANNEL(X) SSLCAUTH(REQUIRED) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)

Configuration



- **Once you have created a Key store for the server to use:**
  - `ALTER QMGR SSLKEYR(<location of keystore>)`
- **Once you have created the certificate for the server to use (MQv8+ only)**
  - `ALTER QMGR CERTLABL(<certificate label>)`
- **To enable TLS on a channel, specify a CipherSpec to use.**
  - `ALTER CHANNEL(<channel name>) CHLTYPE(<channel type>) SSLCIPH(<Cipher string>)`
- **To force clients to connect with a mutual authentication, set the SSLCAUTH to REQUIRED**
  - `ALTER CHANNEL(<channel name>) CHLTYPE(<channel type>) SSLCAUTH(OPTIONAL|REQUIRED)`
- **To set a different certificate to use on a channel (MQv8+ only)**
  - `ALTER CHANNEL(<channel name>) CHLTYPE(<channel type>) CERTLABL(<certificate label>)`

CHANNEL AUTHENTICATION

Details

- **Channel authentication rules are filters that can be applied for incoming connections**
  - Allowlisting – Allow connections based on a filter
  - Blocklisting – Block a connection based on a filter

- **The filters are applied on channels and are applied to all incoming connections for that channel**
  - The filter can be either very specific or generic. (Exact channel name or wildcard)

Details

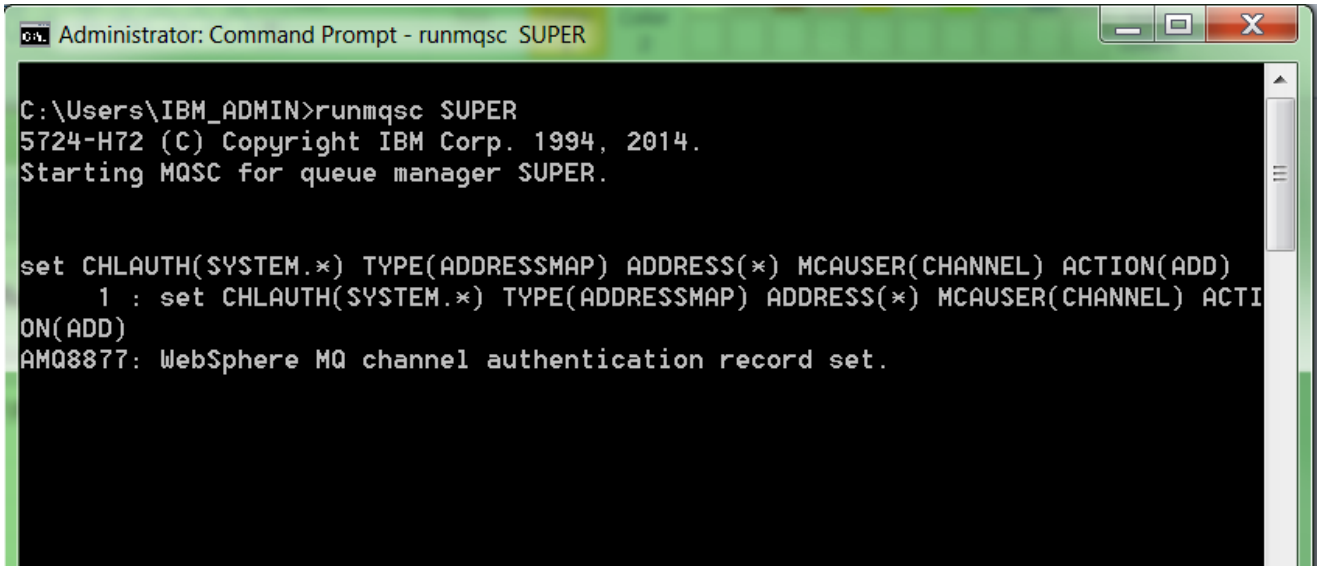
- **There are four types of filters:**
  - TLS Distinguished name (Issuer and Subject)
  - Client User ID name
  - Remote Queue Manager name
  - IP/Hostname

- **For IP/Hostname the connection can be allowed/blocked at the listener or channel**

- **For Client user ID, the userid blocked can be the userid connected with or the final adopted userid**

Configuration

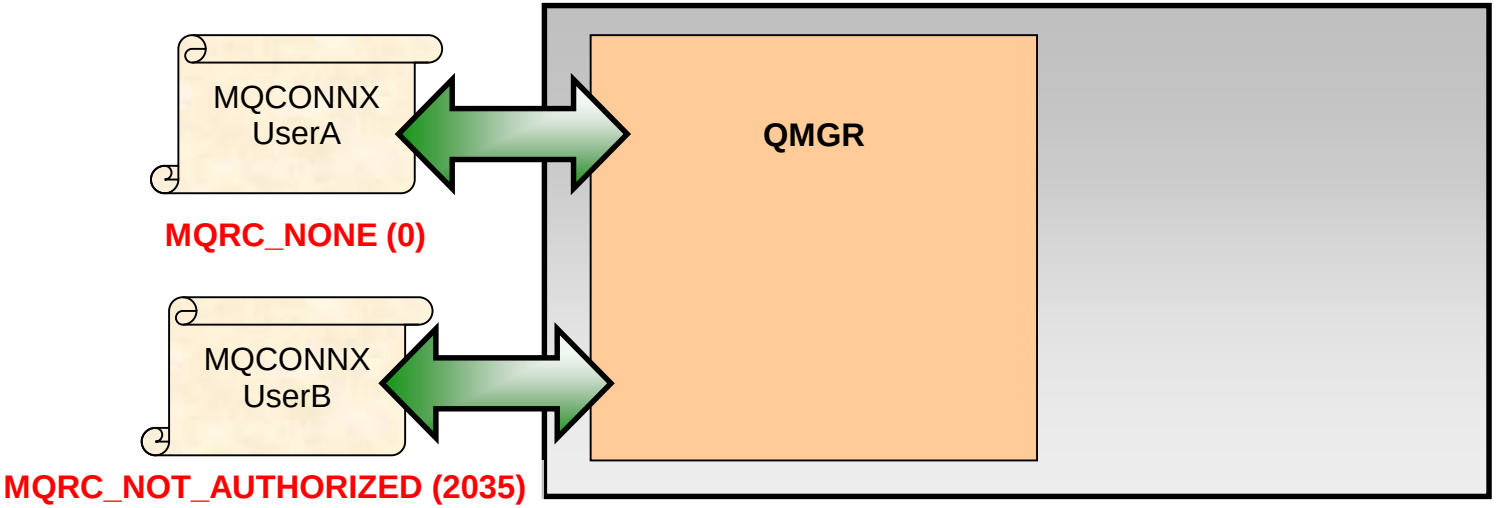
```
SET CHLAUTH(<Channel name>) TYPE(<channel authentication type>)
<extra parameters> ACTION(ADD|REMOVE|REMOVEALL|REPLACE)
```



Configuration

```
SET CHLAUTH(*) TYPE(USERMAP)
               CLNTUSER(*)
               USERSRC(NOACCESS)
               ACTION(ADD)

SET CHLAUTH(*) TYPE(USERMAP)
               CLNTUSER('UserA')
               USERSRC(CHANNEL)
               ACTION(ADD)
```



SECURITY EXITS

Details

- **Security exits are bespoke, customer created exits that are ran during the security checking.**

- **MQ comes with an API that can interact with MQ to provide extra control over a connection.**
  - They allow customers to expand MQ's security to suit their needs.
  - For example a customer could write a security exit to only allow connection to a channel during 08:00 to 17:00.

- **Before MQ v8 they could be used to provide connection authentication functionality.**

- **When executed the security exit will have access to the channel definition, information about the incoming connection and information**
  - It will also have a piece of data passed to it that is set on the channel - SCYDATA

Creation and configuration

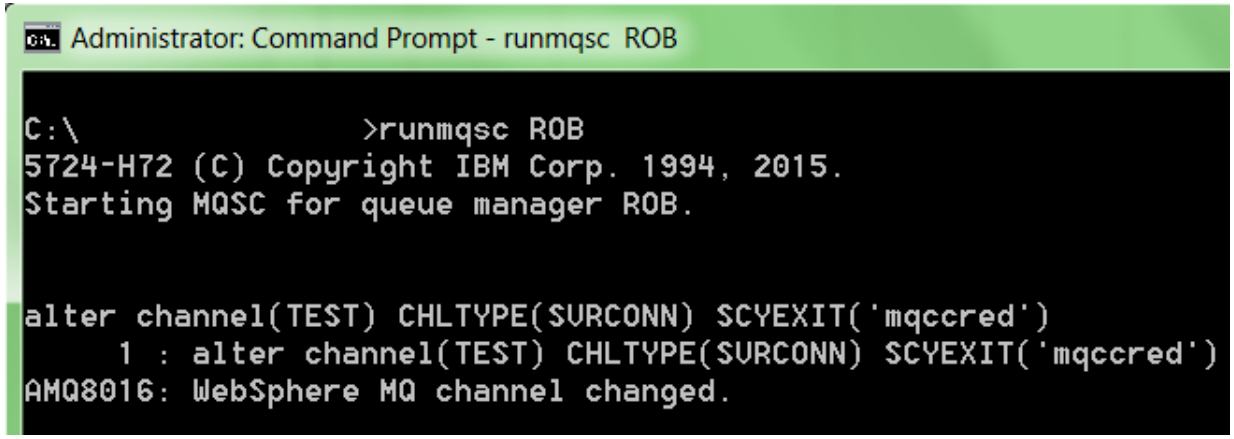
- **To create a Security exit, first create a C file with the following skeleton code.**

```
void MQENTRY MQStart() {;}
void MQENTRY EntryPoint (PMQVOID pChannelExitParms,
                         PMQVOID pChannelDefinition,
                         PMQLONG pDataLength,
                         PMQLONG pAgentBufferLength,
                         PMQVOID pAgentBuffer,
                         PMQLONG pExitBufferLength,
                         PMQPTR  pExitBufferAddr)
{
  PMQCXP pParms = (PMQCXP)pChannelExitParms;
  PMQCD  pChDef = (PMQCD)pChannelDefinition;
  /* Add Security Exit Code Here */
}
```

Creation and configuration

- **Next compile the C file into a dll and place it into:**
  - `<MQ Data Root>/exits/<Installation Name>`

- **With the exit in place you can now edit the channel configuration you want the exit to be invoked on**
  - `ALTER CHANNEL(<channel name>) CHLTYPE(<channel type>)`
    `SCYEXIT(<exit filename without extension>)`
    `SCYDATA(<Data to pass to security exit>)`



AMS

Details

- **AMS stands for Advanced Message Security**
  - It is message level security
  - It is a separate licensable feature - included in MQ Advanced

- **AMS is an end-to-end security model, messages stay signed/encrypted through the whole lifetime of a message**
  - In transit
  - At rest

- **With AMS you can create policies for a queue that describe how messages should be protected when applications put or get messages using that queue name.**
  - Signing
  - Encryption
  - Both

Details

- **AMS does not perform any access control:**
  - Only privacy and integrity protection
  - Should be used with existing access control, authentication, etc
- **Encryption level protection prevents unauthorised users reading message data.**
  - Including MQ administrators.
- **Signing protection prevents messages from being altered.**
- **Signing & Encryption use certificates – Same as TLS.**
- **No application code changes required to use AMS.**

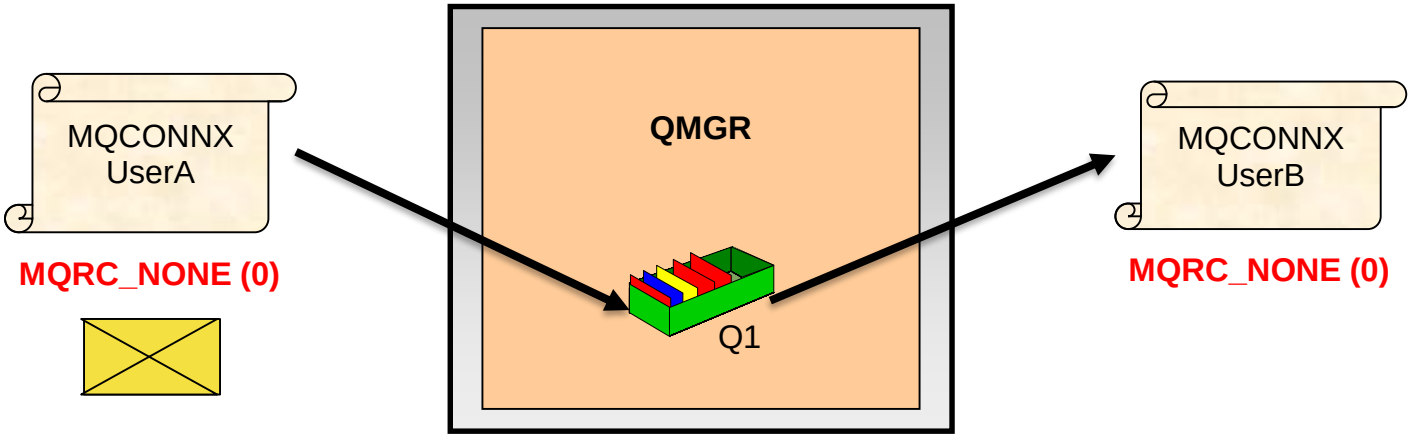
Configuration

- **Differences between AMS & TLS configuration**
  - Both sides must have a certificate
  - Both sides must have exchanged the public certificate
  - The full certificate chain must be present in the key store

- **Policies can be created in explorer, runmqsc or using setmqspl**
  - ```
    setmqspl –m <QM name> -p <Q Name> -s <Signing algorithm>
    -a <Authorised signers> -e <Encryption algorithm> -r <Recipients>
    ```
  - ```
    SET POLICY(<Q NAME>) SIGNALG(<Signing algorithm>)
    ENCALG(<Encryption algorithm>) SIGNER(<Authorised signers>)
    RECIP(<Recipients>) ACTION(ADD|REPLACE|REMOVE)
    ```

Configuration

```
SET POLICY(Q1) SIGNALG(SHA512)
               SIGNER('CN=UserA,O=IBM,C=UK')
               ACTION(ADD)
```



Where can I get more information?

Blog posts tagged with "cloud"

IBM Messaging developerWorks
developer.ibm.com/messaging

IBM Messaging Youtube
https://www.youtube.com/IBMmessagingMedia

LinkedIn
Ibm.biz/ibmmessaging

~~Twitter~~
~~@IBMMessaging~~

~~IBM MQ Facebook~~
~~Facebook.com/IBM-MQ-8304628654/~~

Would you like to take part in IBM MQ Design Research?

- **The IBM MQ team is currently conducting some long term research with our MQ customer base.**
- **With this survey we would like to understand:**
  - Who is interreacting with MQ and what are their responsibilities?
  - Which customers are interested in moving IBM MQ into the cloud?
  - Which customers would like to take part in future research?
- **We estimate the survey should take 4 minutes to complete.**
- **Please note: This survey is for distributed users only.**
- **If you're interested, go to ibm.biz/MQ-Customer-Survey**

Questions & Answers

Notices and disclaimers

Copyright © 2017 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. This document is distributed "as is" without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented
as illustrations of how those customers have used
IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which
IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the
views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions
the customer may need to take to comply with such
laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notices and disclaimers continued

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular, purpose.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services®, Global Technology Services®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli® Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.