IBM
"Wildfire" Workshops

IBM MQ for z/OS– Overview of Resource Security

Lyn Elkins – elkinsc@us.ibm.com
Mitch Johnson – mitchj@us.ibm.com

Wildfire
Technical Hands-On Workshops
IBM zGrowth Team

Agenda

- This session will not cover channel security, covered in detail on other sessions
- What's special on z/OS?
  - Security Switches
- MQ Security Classes
- Connection Security
- Command Security
- Object Security
- Who am I? – What ID is used and the processing options
  - Batch
  - CICS
  - IMS

What's Special?

- On z/OS there are a number of things, but one difference is the security cache and control over that cache
  - On z/OS when security is enabled, the security cache includes all the USERs connected and the MQ resources the user has accessed
  - On distributed, the cache contains group memberships of the users

IBM MQ for z/OS Wildfire Workshop

What's Special? – Part Deux?

- ALTER SECURITY – controls the security scavenger
  - On z/OS the ALTER SECURITY command allows control over the TIMEOUT and INTERVAL
    - The security scavenger runs at predefined INTERVALs. Valid values are:
      - 0 – the scavenger never runs
      - 1-10080 – the interval in minutes between scans
      - 12 – the default
    - The TIMEOUT controls how long entries are kept when they are unused. Valid values are:
      - 0 – if INTERVAL is non-zero all unused entries are purged at each interval
      - 1-10080 – the time in minutes unused entries remain on the cache
      - 54 – the default

What's Special? – Part III?

- Security Switch Profiles are unique to MQ for z/OS
  - The control whether a particular resource, from the subsystem itself to queues, are checked for authorization.
    - To turn off security checking for a queue manager: qmgr.NO.SUBSYS.SECURITY
      - If this setting is used, NO FURTHER SECURITY CHECKS ARE MADE FOR THIS QUEUE MANAGER!
      - This should never be used on a production queue manager, used sparingly on development queue managers
  - To turn on security checking for processes: qmgr.YES.PROCESS.CHECKS

IBM MQ for z/OS Wildfire Workshop

What's Special? – Part III?

- Can be at queue manager or queue sharing group level
  - Specific queue manager switches take precedence over a QSG level.
- Security Switch Format - hlq.option.resource, where
  - hlq = QMGR or QSG
  - option = YES or NO
  - Resource = valid MQ resource type

Queue manager or QSG Control?

Figure 1. Checking for queue manager level security

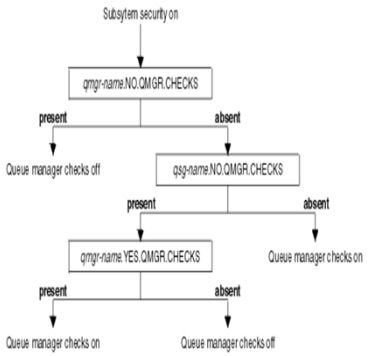
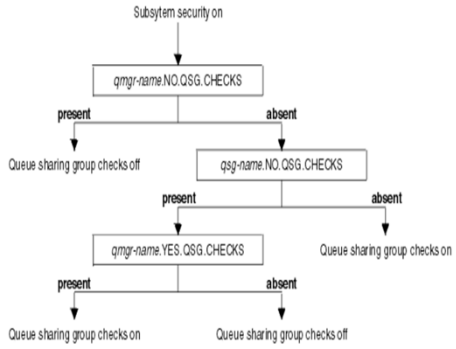


Figure 2. Checking for queue-sharing group level security



IBM MQ for z/OS Wildfire Workshop

Security Switches - Queue manager or QSG Control?

Switch profile name	Type of resource or checking that is controlled
qmgr-name.NO.QMGR.CHECKS	No queue manager level checks for this queue manager
qsg-name.NO.QMGR.CHECKS	No queue manager level checks for this queue-sharing group
qmgr-name.YES.QMGR.CHECKS	Queue manager level checks override for this queue manager
qmgr-name.NO.QSG.CHECKS	No queue-sharing group level checks for this queue manager
qsg-name.NO.QSG.CHECKS	No queue-sharing group level checks for this queue-sharing group
qmgr-name.YES.QSG.CHECKS	Queue-sharing group level checks override for this queue manager

- Unsolicited advice – BE SPECIFIC!

Other Switch controls

Resource	Switch Profile
Connections	qmgr.option.CONNECT.CHECKS
Queues	qmgr.option.QUEUE.CHECKS
Process	qmgr.option.PROCESS.CHECKS
Namelist	qmgr.option.NLIST.CHECKS
Context	qmgr.option.CONTEXT.CHECKS
Alternate User	qmgr.option.ALTERNATE.USER.CHECKS
Commands	qmgr.option.COMMAND.CHECKS
Command resource	qmgr.option.CMD.RESC.CHECKS
Topic	qmgr.option.TOPIC.CHECKS

MQ Security Classes

- Some classes have either an MQ or MX form, MX is used when mixed case support is required.
- One class to rule them – MQADMIN or MXADMIN
  - As the name implies is used primarily for administration
  - Owns a number of profiles, most of the classes just have one
  - Profiles owned:
    - Security Switches
    - RESLEVEL
    - Command Resource
    - Message Level controls
      - Alternate User
      - Context

MQ Security Classes - others

- MQCONN – Connection Security
  - Connection profiles
- MQCMDS – Command Security
  - Command profiles
- MQQUEUE or MXQUEUE – Queue Security
  - Queue profiles
- MQPROC or MXPROC – Process Security
  - Process profiles
- MQNLIST or MXNLIST – Namelist Security
  - Namelist Profiles
- MXTOPIC – Topic Security
  - Topic profiles

MQADMIN - RESLEVEL Profile

- The RESLEVEL profile controls how many IDs are checked for the connection types
  - If there is no RESLEVEL profile:
    - Job and Task IDs are checked for connections to CICS and IMS
    - Job ID is checked for batch
    - Channel User ID and MCA for CHIN
    - WARNING: If you have a generic profile hlq.** it may be used as a RESLEVEL
  - If there is a RESLEVEL profile and the access is:
    - CONTROL or ALTER – No checks are done
    - READ or UPDATE – IDs checked depend on the connection
    - NONE - Both the task ID and address space ID are checked

Command Resource Security

- Command resource security allows control over the user IDs that can define or update MQ resources.
  - For example, you might want the accounting administration team to be able to update or define queues that have an hlq of GL, ACCT but not CORP
  - It is not the same thing as command security
- Profile format – hlq.TYPE.resource_name
  - Where TYPE is an MQ resource or object type, like QUEUE

Command Resource Security Example

- To restrict all users from defining queues based on the GL, ACCT and COPR queue name hlq :
  - RDEFINE MQADMIN QML1.QUEUE.GL* UACC(NONE)
  - RDEFINE MQADMIN QML1.QUEUE.ACCT* UACC(NONE)
  - RDEFINE MQADMIN QML1.QUEUE.CORP* UACC(NONE)
- To add the user ID for the GL administration groups to be able to define and alter queues on queue manager QML1
  - PERMIT QML1.QUEUE.GL* CLASS(MQADMIN) ID(GLADMIN) ACCESS(ALTER)

Message Level Controls - Alternate User ID

- This is security at a message level
- Controlled via a switch profile
- If your applications populate the alternate user ID field in the MQMD and want that ID used for security checking, this MQADMIN/MXADMIN profile is needed.
- Alternate user ID checking is used for:
  - Queues
  - Processes
  - Namelists

Message Level Controls - Context Security

- Context security is at the message level, based on MQMD fields
- Two parts
  - Identity – User ID, Accounting Token, Application Identity token
  - Origin – Putting application type and name, put date and time, application origin data
- Controlled via a switch profile

Connection Security

- The controls 'local' connections to the queue manager
  - Channel security, including CHLAUTH covered elsewhere
- This is application connection security:
  - CICS
    - Controls connections from CICS regions
    - Profile is hlq.CICS
  - IMS
    - Controls connections from IMS regions
    - Profile is hlq.IMS

IBM MQ for z/OS Wildfire Workshop

Connection Security

- Continued:
  - Batch
    - Controls connections from batch jobs, TSO, USS and DB2 stored procedures
    - Profile is hlq.BATCH
  - Channel Initiator
    - Controls connections from the channel initiator address space

Connection Security Example

- To restrict all connections to queue manager QML1:
  - RDEFINE MQCONN QML1.BATCH UACC(NONE)
  - RDEFINE MQCONN QML1.CICS UACC(NONE)
  - RDEFINE MQCONN QML1.IMS UACC(NONE)
  - RDEFINE MQCONN QML1.CHIN UACC(NONE)
- To add the user ID for a CICS region to access queue manager QML1
  - PERMIT QML1.CICS CLASS(MQCONN) ID(MVS1CICS) ACCESS(READ)

Command Security

- Command security is used to control access to commands
  - Format is hlq.MQcommand.objecttype
  - For example, the application owner for Accounts Payable should be allowed to clear any queue that start with AP.** - something that should only be done in emergencies! But regular application personnel should not.
    - The command profile for QML1 to activate control over the CLEAR QLOCAL command is:
      QML1.CLEAR.QLOCAL
    - Then the application owner must be given ALTER access to the queue profile
      PERMIT QML1.QUEUE.AP** CLASS(MQADMIN) ID(APOWNER) ACCESS(ALTER)

Object security

- Dry enough for you yet???

Object Security - Queues

- Queue level security provides different levels of access for individual or groups of queues
- MQQUEUE or MXQUEUE profiles (MX is used for mixed case names)
- Queue security is checked during MQOPEN and the open processing done in an MQPUT1
- Note that unlike distributed, PUT and GET access is not granular
  - Using QALIAS definitions with distinct GET and PUT inhibiting is used when needed
- Profile format is hlq.queue_name
- Queue profiles are generally the most numerous

Queue Security - example

- The generic high-level profile definition for the Accounts Payable queues, this restricts all access to the queues:
  RDEFINE MQQUEUE QML1.AP.** UACC(NONE)
- To give anyone in the APROLES group put and get access:
  PERMIT QML1.AP.** CLASS(MQQUEUE) ID(APROLES) ACCESS(UPDATE)
- To only allow APADMIN to put messages to queue AP.UPDATE.PROVIDER:
  - Define the QALIAS
    DEFINE QALIAS(AP.ADMIN.UPDATE.PROVIDER) PUT(ENABLED) GET(DISABLED) TARGET('AP.UPDATE.PROVIDER')
  - Define the queue profile
    RDEFINE MQQUEUE QML1. ADMIN.UPDATE.PROVIDER UACC(NONE)

IBM MQ for z/OS Wildfire Workshop

Process Security

- Process level security provides or denies access for individual or groups of processes
- MQPROC or MXPROC profiles (MX is used for mixed case names)
- An example of process security that will allow anyone in the APROLES group to execute a process that begins with AP, but no one else
  - Define the process profile
    RDEFINE MQPROC QML1.AP* UACC(NONE)
  - Allow the APROLES ID access
    PERMIT QML1.AP* CLASS(MQPROC) ID(APROLES) ACCESS(READ)

Namelist Security

- Essentially the same as process security.

Topic Security

- WARNING: Topic security always looks for ‘YES’.
  - Unlike most profiles where if the user is not allowed access at the most specific profile there is no further checking; topic security looks to more generic profiles looks for a positive.
- Topic security uses two profiles:
  - hlq.PUBLISH.topicname
  - hlq.SUBSCRIBE.topicname
- The highest level topic, SYSTEM.BASE.TOPIC, should always be secured.
  - RDEFINE MXTOPIC QML1.PUBLISH.SYSTEM.BASE.TOPIC UACC(NONE)
  - RDEFINE MXTOPIC QML1.SUBSCRIBE.SYSTEM.BASE.TOPIC UACC(NONE)

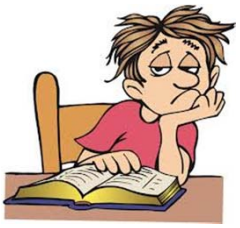
Topic Security - Publications

- Publication security is checked at MQOPEN time
  - Much like QUEUE security
  - For the TOPIC or the QALIAS
  - UPDATE access is required to publish
- No access to queues is necessary

Topic Security - Subscriptions

- Subscriptions require access to both the topic and the queue being used as the target for the subscriptions.
  - If using managed subscriptions, then access to the model queue is required

Who am I?



IBM MQ for z/OS Wildfire Workshop

Who am I?

- Who am I?
  - Not philosophical, but what ID is being used to connect to the queue manager or run the transaction?
    - And the answer is of course, it depends.

Batch User ID used

- The user ID associated with the batch job is used for security checking
  - The connection can be made via an MQCONN or MQCONNX
  - This can be:
    - The TSO user ID
    - The user ID assigned to a batch job by the USER JCL parameter
    - The user ID assigned to a started task by the STARTED class or the started procedures table

CICS ID used

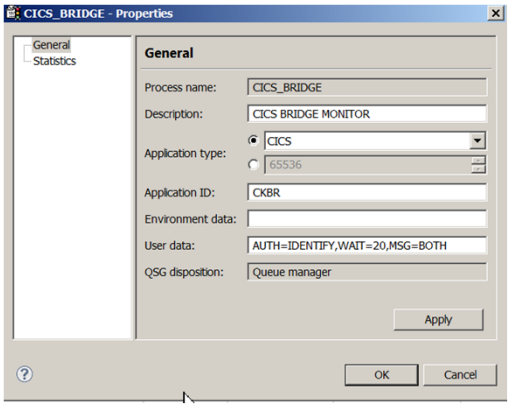
- If MQCONN security is active, the ID used is either the ID associated with the CICS region, or the default ID defined for the CICS region could have a very high level of security
  - Note that a CICS transaction can issue an MQCONNX, but it does not allow additional security checking. If the CSP is sent an environment error is returned.
- If using a CICS triggered transaction the ID used for the trigger monitor transaction (CKTI)
  - The ID associated with the CICS region
  - The default user ID as defined
  - The ID of the user that issued the STARTCKTI command

CICS Bridge User ID used

- MQ CICS bridge security is more granular than triggered transactions.
- It is based on
  - How the bridge monitor program is started
  - What level of checking is specified
- The MQ CICS Bridge is typically triggered, but there is no requirement for this
  - If triggered, the user id used for security checking is based on a special value passed to the CICS Bridge monitor program defined on the MQ process definition

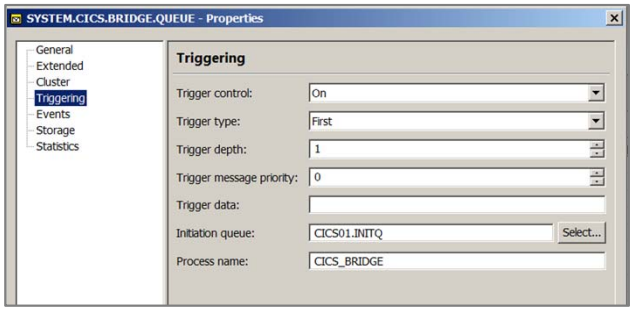
The MQ CICS Bridge User IDs

- The MQ process for the Bridge allows definition showing the AUTH attribute in the user data.



The MQ CICS Bridge – Queue definition

- The sample MQ queue definition for the CICS bridge



MQ CICS Bridge – Starting the bridge manually

- Starting the CICS Bridge from an 'green screen' is simple
- There are six parameters:
  - Q – the name of the queue to be monitored, this defaults to SYSTEM.CICS.BRIDGE.QUEUE
  - AUTH – the authority level, this defaults to LOCAL
  - WAIT – the wait time for additional messages, tis defaults to unlimited
  - MSG – whether messages are to go to the CICS JES log, the master terminal or both. Both is the default
  - PASSTKTA – Defaults to this regions CICS applid. If supplied, this gives the applid to be used for validating the passticket
  - ROUTEMEM – If messages expire, should they be sent to the queue manager dead letter queue. N for no is the default.

The MQ CICS Bridge User IDs

- The AUTH value in the user data can have the following values:
  - LOCAL – the default
    - CICS programs run by the bridge task are started with the CICS DFLTUSER user ID
  - IDENTIFY
    - The user ID from the message descriptor (MQMD) is used, there is no password checking
  - VERIFY_UOW
    - IF MQMD.PutApplType is set to MQAT_NO_CONTEXT
      - It is the same as using LOCAL - the CICS DFLTUSER user ID is used
    - Else
      - The bridge monitor verifies the user ID from the MQMD and the password from the CIH
      - All messages that follow are assumed to be for the same user ID and password.
  - VERIFY_ALL
    - Like VERIFY UOW, except each message is checked individually

IBM MQ for z/OS Wildfire Workshop

The MQ CICS Bridge User IDs

- Warning:
  - The bridge task will run under LOCAL authority when no user ID is passed in the MQMD or password in the MQCIH, even if you started the bridge monitor with a different authentication option

IMS - MQ ID used

- For requests coming from IMS regions to MQ, the ID the of the IMS region is used when checking resource security

IMS Bridge IDs used

- Security for IMS Bridge transaction is based on the setting of the SECURE OTMA command
  - NONE
    - No security checks are made
  - CHECK
    - The user id from the MQMD is used by IMS for security checking
    - An ACEE (Accessor Environment Element) is built in the IMS control region
  - FULL
    - Like the CHECK, an ACEE is built in both the control and dependent region
  - PROFILE
    - Like CHECK, the building of the ACEE in the dependent region is based on the SecurityScope field of the IIH header passed on the request

Queue Security checking done by the Bridge

- IMS Bridge Queue – no checking is done for the MQGET
- Putting a reply or a report message (COA, COD, etc.) – the user ID from the MQMD is used to check authority for putting to the queue
- Putting a message to the dead letter queue – no checking is done

Finally – useful commands

- ESM command to refresh security changes (RACF example):

  RACF SETROPTS RACLIST(classname) REFRESH

- MQ Command REFRESH SECURITY
  - Used on both z/OS and distributed
    - Clears the security cache so that changes will be loaded and used
    - Options vary and the documentation is messy

Advice & Summary

- Security in MQ can get very complex
  - Use groups and generic profiles to reduce the number of definitions
  - Good naming standards are important
- Production resources should always be secured
- Where possible – run all new security implementation in warning mode first