SHARE in Orlando 2015

# Why Shouldn't I Be Able To Open This Queue? MQ and CICS Security Topics - 16544

*Mitch Johnson – mitchj@us.ibm.com*

#SHAREorg

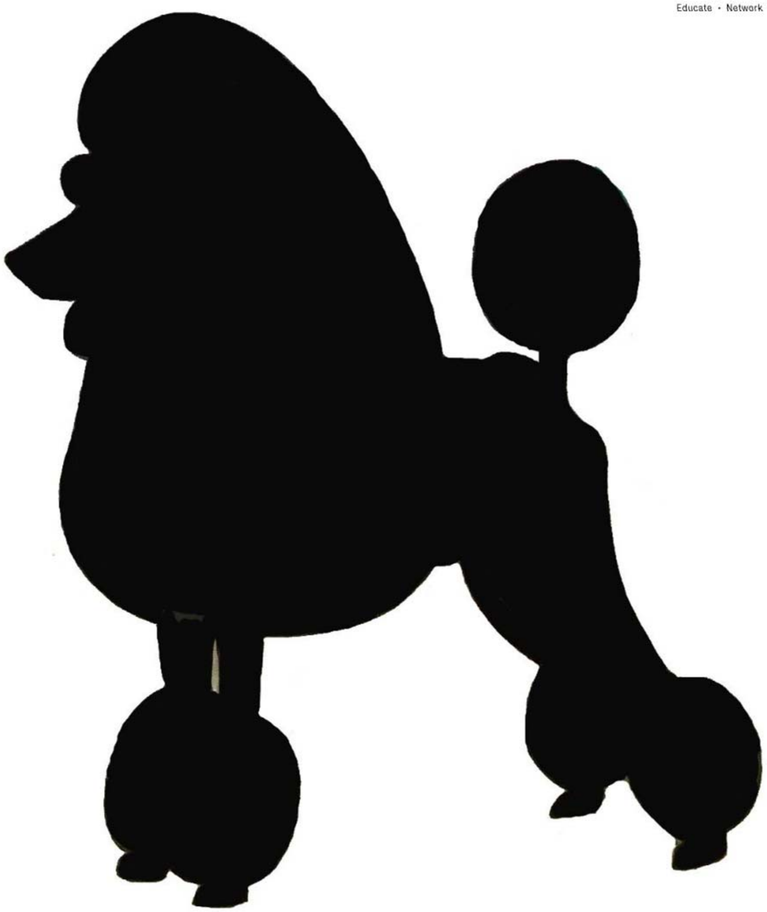
# The code for session 16544

# Agenda

- Why is this important?
- Warning - This is written from an MQ perspective
- MQ Security overview
  - MQ Profiles
    - Switch profiles
  - Connection security
  - Queue security
- CICS-MQ adapter security
- Triggering
- CICS Bridge
- Other Security Info
- Summary

# Why is this important?

- New security vulnerabilities exposed regularly
  - Heartbleed
  - Poodle
  - ……
- In this presentation we will not be talking about TLS – channel security – just MQ and CICS
- Finally, in addition to the vulnerabilities the real reason most people are interested:
  - Our auditors are forcing us!



# Why is this important? Notes

- Most of the security vulnerabilities that have made the news are about networking and cipher spec related issues. Those do not really apply to MQ and CICS, because the connections are local.  What has happened is an increased focus on security from head to toe.
- Over the years many enterprises have been less than stringent about the locally attached subsystems, having faith in the belief that the external applications were doing as they promised – enforcing strong authentication and security protocols.  That faith has been somewhat shaken, especially when so many of the new applications that want to ‘see’ and use data of record systems are built on OpenSource tools.

# Basic MQ for z/OS security

*The 50,000 foot view*

# MQ Security Classes and Profiles

- To secure any MQ resource the MQADMIN or MXADMIN class must be activated:
  - MXADMIN is used for mixed case profiles
  - We see some mixed case, but it is still primarily upper case checking
- Security can be at the queue manager or queue sharing group level – or a combination
  - For a complete description of the order of precedence, please see http://www-01.ibm.com/support/knowledgecenter/SSFKSJ_8.0.0/com.ibm.mq.sec.doc/q011490_.htm

# MQ Switch Profiles

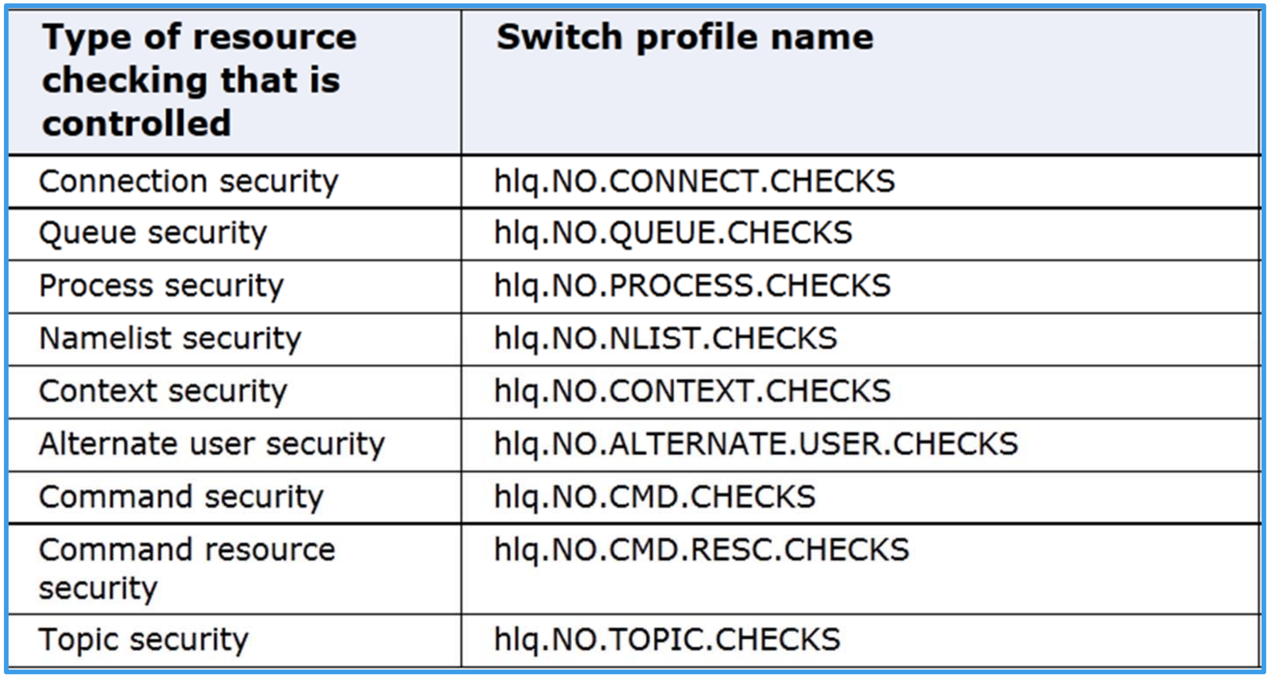
- Switch profiles provide great flexibility in securing resources
- If a ‘NO’ switch is found for any resource, then security checking is not performed for that resource.
- For example, to turn security completely off for a test queue manager:

  ```
  RDEFINE MQADMIN QML1.NO.SUBSYS.SECURITY OWNER(SYS1)
  ```

- This should never be done for a production queue manager
  - Especially if the auditors are watching

# MQ Switch Profiles Continued

- The following switch profiles may be defined for MQ objects and operations:



| Type of resource checking that is controlled | Switch profile name |
|---|---|
| Connection security | hlq.NO.CONNECT.CHECKS |
| Queue security | hlq.NO.QUEUE.CHECKS |
| Process security | hlq.NO.PROCESS.CHECKS |
| Namelist security | hlq.NO.NLIST.CHECKS |
| Context security | hlq.NO.CONTEXT.CHECKS |
| Alternate user security | hlq.NO.ALTERNATE.USER.CHECKS |
| Command security | hlq.NO.CMD.CHECKS |
| Command resource security | hlq.NO.CMD.RESC.CHECKS |
| Topic security | hlq.NO.TOPIC.CHECKS |

# MQ Connection Security

- Connection security is straightforward
- If the MQCONN class is active and connection security has not been turned off:
  - The user ID associated with the CICS region is checked for read access to the MQ CICS profile
  - The profile has the format:
    hlq.CICS – where hlq is the queue manager SSID (checked first) or the queue sharing group.
  - Example RACF commands to allow CICS user ID CICSUSER access to queue manager QML1:
    - To restrict all CICS connections to queue manager QML1:
      ```
      RDEFINE MQCONN QML1.CICS UACC(NONE)
      ```
    - To allow any CICS region that uses the ID CICSUSER to connect:
      ```
      PERMIT QML1.CICS CLASS(MQCONN) ID(CICSUSER)
      ACCESS(READ)
      ```

# MQ Connection Security - notes

- MQ CICS Connection security can be used to restrict a CICS region to a specific queue manager or queue sharing group.
- One issue with the connection is that the user id from the connection between MQ and CICS is often used for access to the other MQ resources.
- The ID used (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
- Pointers to Knowledge Center information:
  - Connection security
    http://www-01.ibm.com/support/knowledgecenter/SSFKSJ_8.0.0/com.ibm.mq.sec.doc/q011540_.htm
  - CICS connection security
    http://www-01.ibm.com/support/knowledgecenter/SSFKSJ_8.0.0/com.ibm.mq.sec.doc/q011560_.htm

# Resource Security

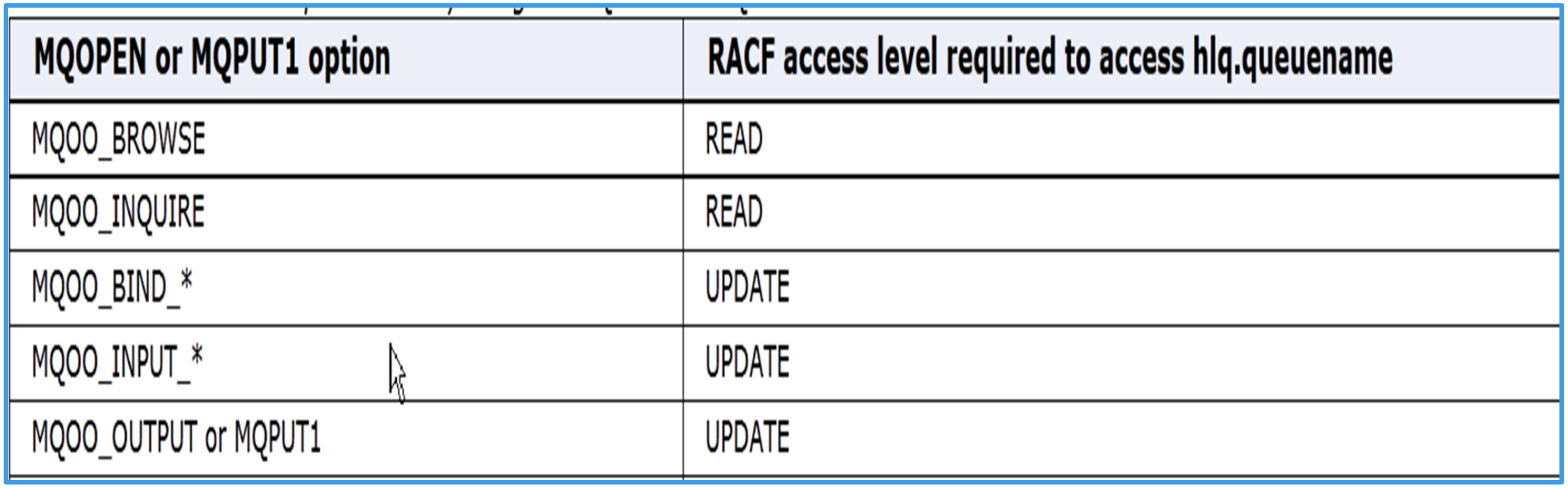
- From an MQ and CICS perspective, the important resources to secure are
  - Queues
    - Typically queues are owned by applications
    - The user ID associated with the transaction accessing the queue must have the proper access
  - Topics
    - Treated like queues

# MQ Object Security - Queues

- The most typical concern is securing the queue and topic objects
- To secure a queue, or set of queues for an application:
  - Define profiles in the MQQUEUE (or MXQUEUE) class
  - The profile is named hlq.queuename where:
    - ‘hlq’ is the queue manager or queue sharing group name
    - Queue name is either a specific or generic queue name to be protected

# MQ Object Security - Queues

- The following excerpt shows the queue access type and the corresponding RACF Level requirements
  - Note there are other options, but that is beyond the scope of this session.



| MQOPEN or MQPUT1 option | RACF access level required to access hlq.queuename |
|---|---|
| MQOO_BROWSE | READ |
| MQOO_INQUIRE | READ |
| MQOO_BIND_* | UPDATE |
| MQOO_INPUT_* | UPDATE |
| MQOO_OUTPUT or MQPUT1 | UPDATE |

# MQ Object Security - Queues

- Example:
  - Define the MQQUEUE profile for the queue
    - RDEFINE MQQUEUE QML1.APP1.REPLY OWNER(MQADMN)
  - Permit anyone in the APP2 group to browse the queue
    - PERMIT QML1.APP1.REPLY  CLASS(MQQUEUE) ID(APP2) ACC(READ)
  - Permit anyone in the APP1 group to PUT and GET from the queue
    - PERMIT QML1.APP1.REPLY  CLASS(MQQUEUE) ID(APP1) ACC(UPDATE)

# MQ Queue Knowledge Center pointers

- Queue Security
  - http://www-01.ibm.com/support/knowledgecenter/SSFKSJ_8.0.0/com.ibm.mq.sec.doc/q011590_.htm

# MQ Object Security - Queues

- Because MQPUT and GET are not granular on z/OS, alias queues may be used when additional control is needed
  - Example, allow APP1 to only PUT to the reply queue and APP2 to only GET from the reply queue for queue manager QML1
  - Exclude everyone from access the base queue:
    - MQ - DEFINE QLOCAL(QML1.APP1.REPLY) ....
    - RACF  - PERMIT QML1.QML1.APP1.REPLY CLASS(MQQUEUE) UACC(NONE)
  - Definitions used so that APP2 ID is only allowed to GET from the reply
    - MQ - DEFINE QALIAS(QML1.APP2.REPLY) PUT(DISABLED) TARGET(QML1.APP1.REPLY) TARGTYPE(QUEUE)
    - RACF - PERMIT QML1.QML1.APP2.REPLY  CLASS(MQQUEUE) ID(APP2) ACC(UPDATE)
  - Definitions so that APP1 is only allowed to PUT to the reply queue
    - MQ - DEFINE QALIAS(QML1.APP1.REPLY2) GET(DISABLED) TARGET(QML1.APP1.REPLY) TARGTYPE(QUEUE)
    - PERMIT QML1.QML1.APP1.REPLY2  CLASS(MQQUEUE) ID(APP1) ACC(UPDATE

# MQ and CICS - Now a look at the user IDs used

# General User IDs used

- From the CICS Knowledge Center
- Adapter Tasks Associated with a terminal
  - The user ID associated with the CICS-WebSphere MQ adapter is that of the WebSphere® MQ-supplied task initiator transaction, CKTI.
  - For terminal tasks where a user has not signed on, the user ID is the CICS user ID associated with the terminal and is either:
    - The default CICS user ID as specified on the CICS parameter DFLTUSER SIT
    - A preset security user ID specified on the terminal definition
- For non-terminal tasks:
  - An EXEC CICS ASSIGN command is used to get a user id.
  - If that does not work the adapter tries to get the user ID using EXEC CICS INQUIRE TASK.
  - If security is active in CICS, and the non-terminal attached transaction is defined with CMDSEC(YES), the CICS adapter passes a user ID of blanks to WebSphere MQ.

- These days we fairly rarely see MQ CICS transactions started from a terminal. More often the transactions are either long running CICS transactions (often started via PLT), triggered, or started from the CICS bridge.
- The ID associated with a triggered transaction is often the default ID.  That ID may have too high a level of security for comfort.

# MQ enabled CICS Programs

- Security checking is done at MQOPEN time
- MQ uses the ID of the transaction being executed at the time of the MQOPEN

# MQ Triggering

- Triggering is one of the more common ways CICS transactions are initiated by MQ
- These transactions are initiated automatically when a trigger event occurs, as long as the CICS trigger monitor (CKTI) is active and the queue has been defined correctly.
  - Trigger events are typically defined :
    - First – when the queue depth goes from 0 to something greater
    - Every – each message put to the queue has a trigger event
- This is simply a ‘special’ type of an adapter task
  - The ID used can be:
    - The ID associated with the CICS region
    - The default user ID as defined
    - The ID of the user that issued the STARTCKTI command

# User ID used for triggering  - Notes

- There are other types of triggers, but First and Every are the most common.
  - For more information on MQ Triggering please see: [http://www-01.ibm.com/support/knowledgecenter/SSFKSJ_8.0.0/com.ibm.mq.dev.doc/q026910_.htm](http://www-01.ibm.com/support/knowledgecenter/SSFKSJ_8.0.0/com.ibm.mq.dev.doc/q026910_.htm)
- A suggestion:
  - It would be beneficial if a user ID and password could be associated with the STARTCKTI transaction, control would be easier and more straightforward
  - If others feel like this, please open a CICS RFE to provide this capability

# The CICS Bridge

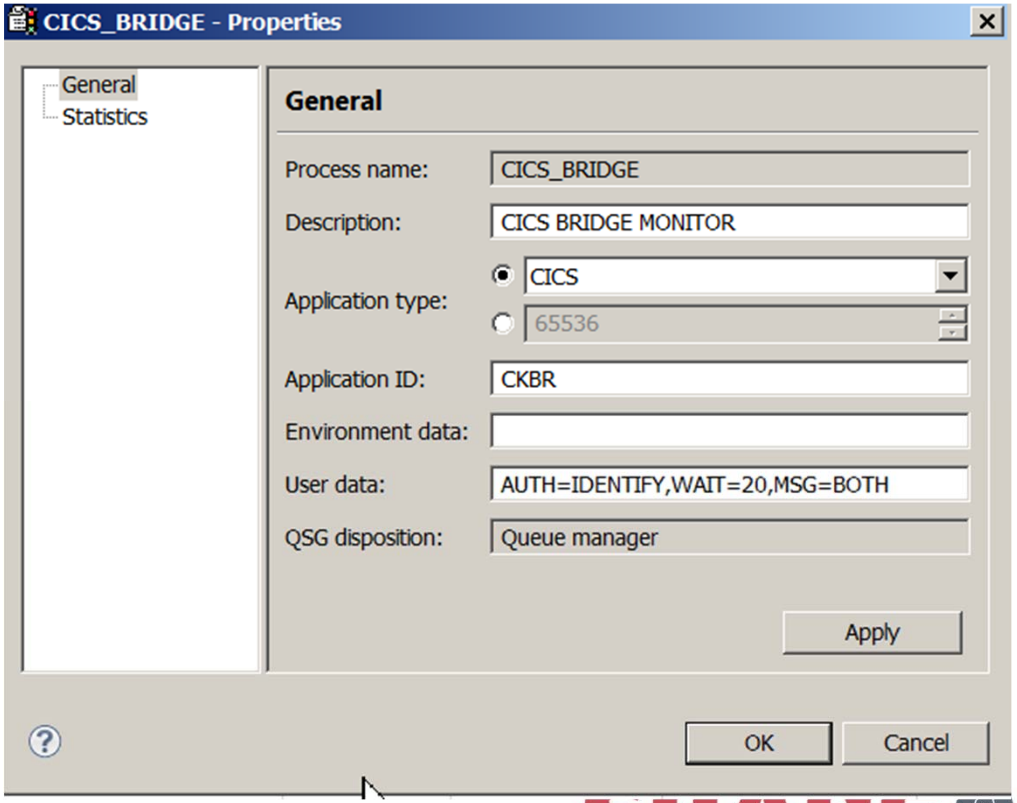
- The CICS Bridge is the next most popular way of initiating CICS transactions via MQ messages
- Transaction CKBR:
  - Is the standard CICS bridge monitor task, it is associated with program
  - Retrieves messages from the bridge defined queue
  - Issues a CICS LINK to the program specified in the message body or in the special MQ CIH header prepended to the message
  - The message body is sent to the program as a COMMAREA or a container

# The MQ CICS Bridge Security

- MQ CICS bridge security is more granular than typical 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

- Sample MQ process definition showing the AUTH attribute in the user data.



General

Process name: CICS_BRIDGE

Description: CICS BRIDGE MONITOR

Application type: ● CICS ○ 65536

Application ID: CKBR

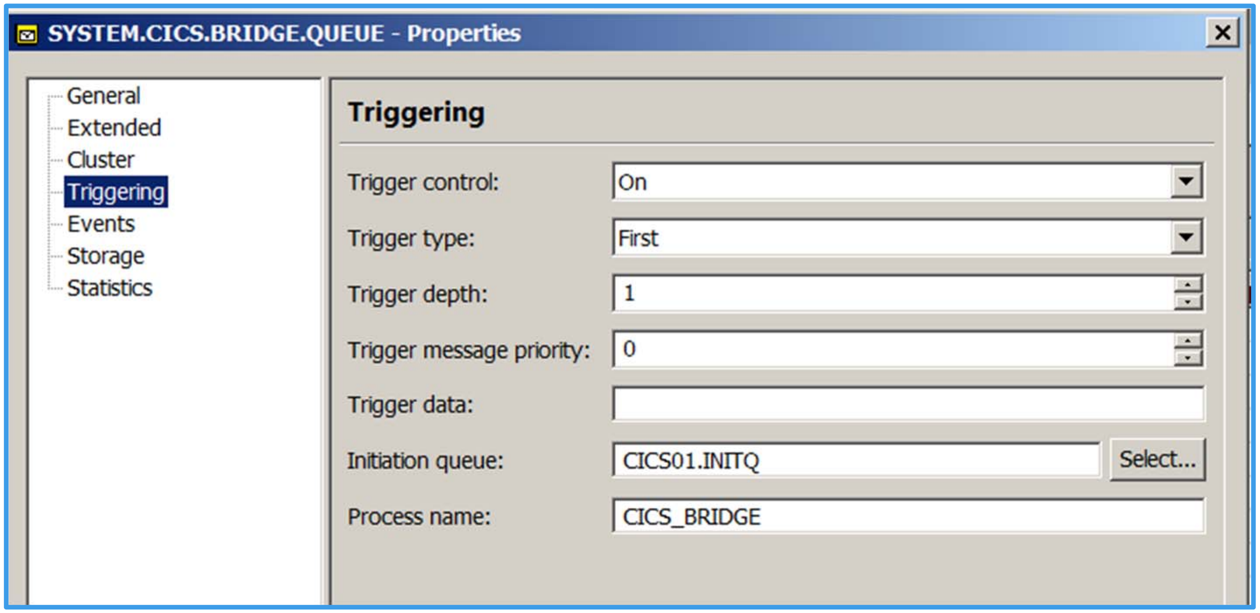
Environment data:

User data: AUTH=IDENTIFY,WAIT=20,MSG=BOTH

QSG disposition: Queue manager

# The MQ CICS Bridge – Queue definition

- The sample MQ queue definition for the CICS bridge



Triggering

Trigger control: On

Trigger type: First

Trigger depth: 1

Trigger message priority: 0

Trigger data:

Initiation queue: CICS01.INITQ

Process name: 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.

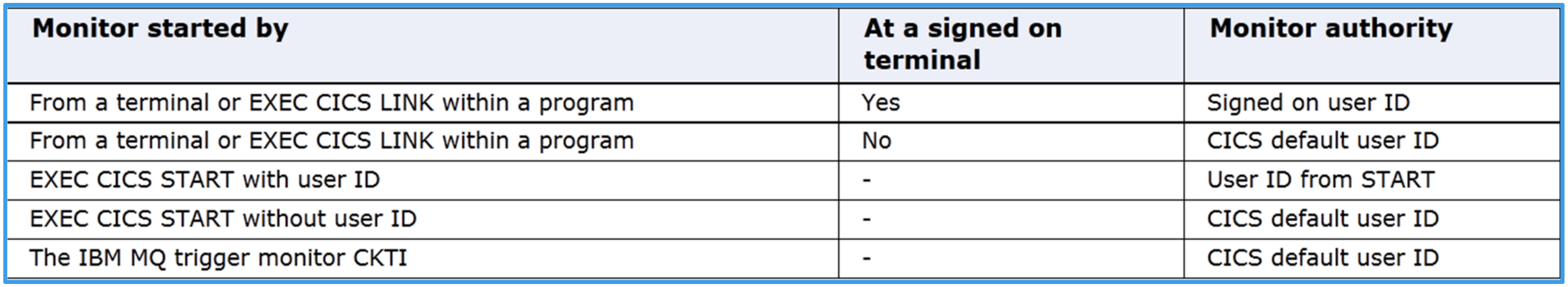
# The MQ CICS Bridge User IDs

- 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

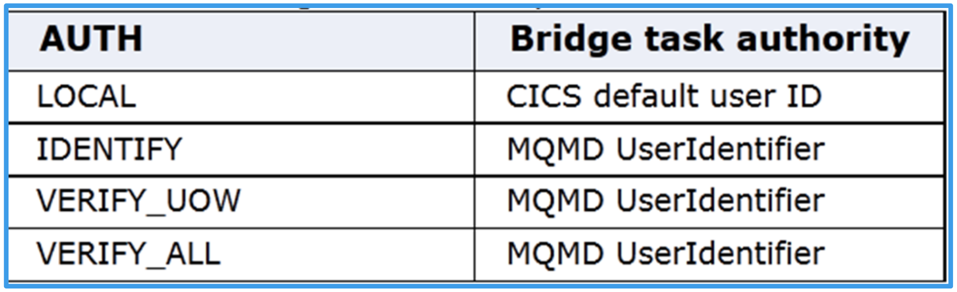
# The MQ CICS Bridge User IDs

- 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

# The MQ CICS Bridge - User IDs



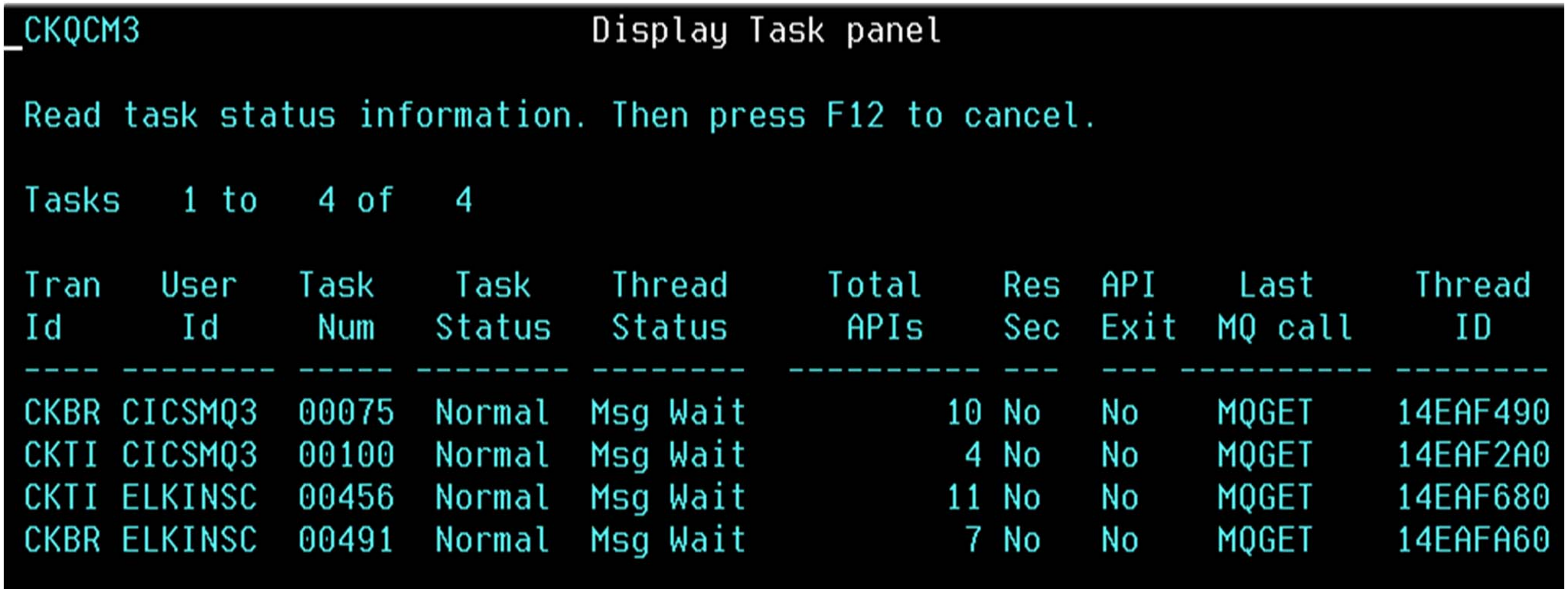
| Monitor started by | At a signed on terminal | Monitor authority |
|---|---|---|
| From a terminal or EXEC CICS LINK within a program | Yes | Signed on user ID |
| From a terminal or EXEC CICS LINK within a program | No | CICS default user ID |
| EXEC CICS START with user ID | - | User ID from START |
| EXEC CICS START without user ID | - | CICS default user ID |
| The IBM MQ trigger monitor CKTI | - | CICS default user ID |



| AUTH | Bridge task authority |
|---|---|
| LOCAL | CICS default user ID |
| IDENTIFY | MQMD UserIdentifier |
| VERIFY_UOW | MQMD UserIdentifier |
| VERIFY_ALL | MQMD UserIdentifier |

# Wondering what user id is being used?

- The CKQC transaction will list the tasks and the user ID associated as shown



```
_CKQCM3                        Display Task panel

 Read task status information. Then press F12 to cancel.

 Tasks    1 to    4 of    4

 Tran    User    Task     Task      Thread        Total     Res  API    Last       Thread
 Id       Id      Num    Status     Status         APIs     Sec  Exit  MQ call       ID
 ---- -------- ----- -------- --------   ---------- ---  --- ---------- --------
 CKBR CICSMQ3  00075  Normal   Msg Wait           10 No   No    MQGET     14EAF490
 CKTI CICSMQ3  00100  Normal   Msg Wait            4 No   No    MQGET     14EAF2A0
 CKTI ELKINSC  00456  Normal   Msg Wait           11 No   No    MQGET     14EAF680
 CKBR ELKINSC  00491  Normal   Msg Wait            7 No   No    MQGET     14EAFA60
```

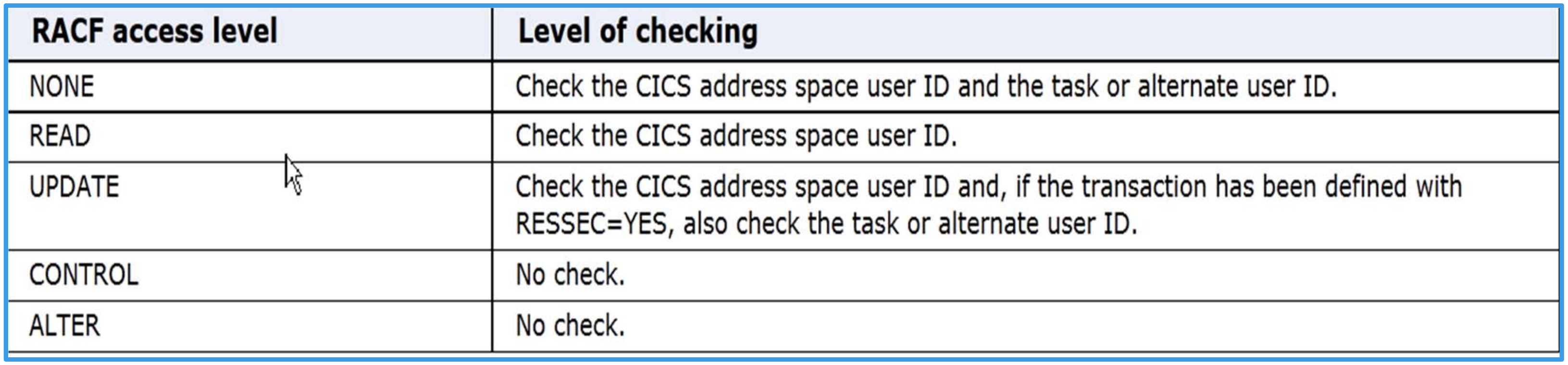
# Other MQ CICS Security considerations

# RESLEVEL and CICS connections

- RESLEVEL profiles control how many user IDs are checked with a CICS application tries to access an MQ object.
- The profile has the following format:
  hlq.RESLEVEL
- **WARNINGS:**
  - RESLEVEL is a very powerful option; it can cause the bypassing of all resource security checks for a particular connection.
  - Using the RESLEVEL profile means that normal security audit records are not taken. For example, if you put UAUDIT on a user, the access to the hlq.RESLEVEL profile in MQADMIN is not audited.
  - If you use the RACF WARNING option on the hlq.RESLEVEL profile, no RACF warning messages are produced for profiles in the RESLEVEL class.

# RESLEVEL and CICS connections - continued

- For CICS two IDs are checked by default:
  - The address space ID
  - The user ID associated with the transaction
- The access level controls the IDs checked as shown in this table:



| RACF access level | Level of checking |
|---|---|
| NONE | Check the CICS address space user ID and the task or alternate user ID. |
| READ | Check the CICS address space user ID. |
| UPDATE | Check the CICS address space user ID and, if the transaction has been defined with RESSEC=YES, also check the task or alternate user ID. |
| CONTROL | No check. |
| ALTER | No check. |

# RESLEVEL and CICS connections

- For additional information please see:

- http://www-01.ibm.com/support/knowledgecenter/SSFKSJ_8.0.0/com.ibm.mq.sec.doc/q011790_.htm

# And if you need encryption at rest

- Normally messages at rest (in a queue) are not encrypted
- If the data contained is sensitive,
  - Advanced Message Security is the solution
    - Messages are encrypted as they are written and decrypted when retrieved
    - Transparent to the application

# The END

- Many thanks to:
  - Mitch Johnson – mitchj@us.ibm.com
  - Shalawn King – shalawn@us.ibm.com
  - Kenishia Calloway - kenishia@us.ibm.com
- Any questions?

# The code for session 16544