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

Lyn Elkins – elkinsc@us.ibm.com
The code for session 16544

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
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!
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/SSFKJS_8.0.0/com.ibm.mq.sec.doc/q011490_.htm](http://www-01.ibm.com/support/knowledgecenter/SSFKJS_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.
The following switch profiles may be defined for MQ objects and operations:

<table>
<thead>
<tr>
<th>Type of resource checking that is controlled</th>
<th>Switch profile name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection security</td>
<td>hlq.NO.CONNECT.CHECKS</td>
</tr>
<tr>
<td>Queue security</td>
<td>hlq.NO.QUEUE.CHECKS</td>
</tr>
<tr>
<td>Process security</td>
<td>hlq.NO.PROCESS.CHECKS</td>
</tr>
<tr>
<td>Namelist security</td>
<td>hlq.NO.NLIST.CHECKS</td>
</tr>
<tr>
<td>Context security</td>
<td>hlq.NO.CONTEXT.CHECKS</td>
</tr>
<tr>
<td>Alternate user security</td>
<td>hlq.NO.ALTERNATE.USER.CHECKS</td>
</tr>
<tr>
<td>Command security</td>
<td>hlq.NO.CMD.CHECKS</td>
</tr>
<tr>
<td>Command resource security</td>
<td>hlq.NO.CMD.RESC.CHECKS</td>
</tr>
<tr>
<td>Topic security</td>
<td>hlq.NO.TOPIC.CHECKS</td>
</tr>
</tbody>
</table>
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)`
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.

<table>
<thead>
<tr>
<th>MQOPEN or MQPUT1 option</th>
<th>RACF access level required to access hlq.queue.name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQOO_BROWSE</td>
<td>READ</td>
</tr>
<tr>
<td>MQOO_INQUIRE</td>
<td>READ</td>
</tr>
<tr>
<td>MQOO_BIND_*</td>
<td>UPDATE</td>
</tr>
<tr>
<td>MQOO_INPUT_*</td>
<td>UPDATE</td>
</tr>
<tr>
<td>MQOO_OUTPUT or MQPUT1</td>
<td>UPDATE</td>
</tr>
</tbody>
</table>
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 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.
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.

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
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.
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, this 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 region’s 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

<table>
<thead>
<tr>
<th>Monitor started by</th>
<th>At a signed on terminal</th>
<th>Monitor authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>From a terminal or EXEC CICS LINK within a program</td>
<td>Yes</td>
<td>Signed on user ID</td>
</tr>
<tr>
<td>From a terminal or EXEC CICS LINK within a program</td>
<td>No</td>
<td>CICS default user ID</td>
</tr>
<tr>
<td>EXEC CICS START with user ID</td>
<td>-</td>
<td>User ID from START</td>
</tr>
<tr>
<td>EXEC CICS START without user ID</td>
<td>-</td>
<td>CICS default user ID</td>
</tr>
<tr>
<td>The IBM MQ trigger monitor CkTI</td>
<td>-</td>
<td>CICS default user ID</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AUTH</th>
<th>Bridge task authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCAL</td>
<td>CICS default user ID</td>
</tr>
<tr>
<td>IDENTIFY</td>
<td>MQMD UserIdentifier</td>
</tr>
<tr>
<td>VERIFY_UOW</td>
<td>MQMD UserIdentifier</td>
</tr>
<tr>
<td>VERIFY_ALL</td>
<td>MQMD UserIdentifier</td>
</tr>
</tbody>
</table>
Wondering what user id is being used?

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

<table>
<thead>
<tr>
<th>Tran Id</th>
<th>User Id</th>
<th>Task Num</th>
<th>Task Status</th>
<th>Thread Status</th>
<th>Total APIs</th>
<th>Res Sec</th>
<th>API Exit</th>
<th>Last MQ call</th>
<th>Thread ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CKBR</td>
<td>CICSM03</td>
<td>00075</td>
<td>Normal</td>
<td>Msg Wait</td>
<td>10</td>
<td>No</td>
<td>No</td>
<td>MQGET</td>
<td>14EAF490</td>
</tr>
<tr>
<td>CKTI</td>
<td>CICSM03</td>
<td>00100</td>
<td>Normal</td>
<td>Msg Wait</td>
<td>4</td>
<td>No</td>
<td>No</td>
<td>MQGET</td>
<td>14EAF2A0</td>
</tr>
<tr>
<td>CKTI</td>
<td>ELKINSC</td>
<td>00456</td>
<td>Normal</td>
<td>Msg Wait</td>
<td>11</td>
<td>No</td>
<td>No</td>
<td>MQGET</td>
<td>14EAF680</td>
</tr>
<tr>
<td>CKBR</td>
<td>ELKINSC</td>
<td>00491</td>
<td>Normal</td>
<td>Msg Wait</td>
<td>7</td>
<td>No</td>
<td>No</td>
<td>MQGET</td>
<td>14EAFAA0</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>RACF access level</th>
<th>Level of checking</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>Check the CICS address space user ID and the task or alternate user ID.</td>
</tr>
<tr>
<td>READ</td>
<td>Check the CICS address space user ID.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>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.</td>
</tr>
<tr>
<td>CONTROL</td>
<td>No check.</td>
</tr>
<tr>
<td>ALTER</td>
<td>No check.</td>
</tr>
</tbody>
</table>
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

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Complete your session evaluations online at www.SHARE.org/Seattle-Eval