Connecting to IIB WebUI Using Queue Based Authentication Step-by-Step

Queue-based authentication allows Webui users to be authenticated via MQ, which provides more robust layer of security. In order to use queue-based authentication, you need to make sure that a queue manager is associated with the Integration node and that you have defined a local system user(s). After you created a system level user(s), you should assign it to a local group(s) also created at OS level. You will use this local group to configure the MQ permissions using the setmqaut command.

You can also use LDAP with queue-based authentication. However, this is beyond the scope of this article, and IIB will not provide instructions on how to do so. If you would like to implement the LDAP configuration, you should contact your LDAP admin for further assistance.

*** *Do not use keywords like iibadmingroup or iibadmin or iibusersgrp. I have tried the iibadmingroup while creating this tutorial, and it did not work for me*.

PART I: Administrative Tasks at OS Level

1. Creating system groups and users at OS level (meaning on your local machine).

Please note that these users are not the webadmin users. They are just dummy users whose privileges will be inherited by the webadmin users you will create later or feed via LDAP. If you don't want to create these dummy users, you can use your own system user and group. This is how the product is designed to work.

In this example, I am creating 3 local users:
iibadm ==> for administrators
iibdev ==> for developers
iibusers ==> for standard users
Also create 3 local groups:
inodeadmgrp ==> for administrator group
inodedevgrp ==> for developers group
stdusersgrp ==> for standard users group

==(OPTIONAL) Delete users and groups if already existents====
sudo userdel --remove iibadm
sudo userdel --remove iibdev
sudo userdel --remove iibusers
sudo groupdel inodeadmgrp
sudo groupdel inodedevgrp
sudo groupdel stdusersgrp

====create users and groups=======
sudo useradd iibadm
sudo useradd iibdev
sudo useradd iibusers
sudo groupadd inodeadmgrp
sudo groupadd inodedevgrp
sudo groupadd stdusersgrp

sudo passwd iibadm
sudo passwd iibdev
sudo passwd iibusers

====Add users to their respective groups=======
sudo usermod -a -G inodeadmgrp iibadm

sudo usermod -a -G inodedevgrp iibdev
sudo usermod -a -G stdusersgrp iibusers
====check that users are in their respective groups========
id -a iibadm
uid=1014(iibadm) gid=1016(iibadm) groups=1016(iibadm),1019(inodeadmgrp)
id -a iibdev
uid=1015(iibdev) gid=1017(iibdev) groups=1017(iibdev),1020(inodedevgrp)
id -a iibusers
uid=1016(iibusers) gid=1018(iibusers) groups=1018(iibusers),1022(stdusersgrp)

PART II: Creating Authorization for MQ Base Authentication

1. Creating the default system queues

The default system queues are no longer created by default. However, a script iib_createqueues.sh is provided in the /opt/IBM/IIB.0.0.0x/server/sample/wmq to help you create those system queues easily. You should always first check and make sure that these system queues are not already created.

To execute the script, run
./iib_createqueues.sh YOURQUEUEMANAGER PrimaryUNIXSystemLevelGroup

Please, note that PrimaryUNIXSystemLevelGroup can be any UNIX system group created. In our case, this group can be inodeadmgrp, or inodeadmgrp, or any dummy group.

These system queues can also be created manually via the MQ command prompt.

2. Setting Up Role-Based Security

If you would like to know how to create permission for multiple EGs, click the link below:
https://www.ibm.com/support/knowledgecenter/en/SSMKHH_10.0.0/com.ibm.etools.mft.doc/bp43640_.htm

Run the following MQ control commands to grant full administrative access to all broker resources for the group inodeadmgrp.
setmqaut -m YOURQUEUEMANAGER -t qmgr -g inodeadmgrp +connect +inq
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.DEPLOY.QUEUE -t queue -g inodeadmgrp +put
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.DEPLOY.REPLY -t queue -g inodeadmgrp +put +get
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.AUTH -t queue -g inodeadmgrp +inq +put +set
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.DC.AUTH -t queue -g inodeadmgrp +inq +set
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.WEBADMIN.SUBSCRIPTION -t queue -g inodeadmgrp +put +get
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.MB.TOPIC -t topic -g inodeadmgrp +sub +pub
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.AUTH.EG -t queue -g inodeadmgrp +inq +put +set

PS: Replace EG by the name of your EG. You should have multiple lines of of the **setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.AUTH.EG -t queue -g inodeadmgrp +inq +put +set** if you have multiple EGs or use the wild card (*) as **SYSTEM.BROKER.AUTH.***

**Verify your configuration

dmpmqaut -m YOURQUEUEMANAGER -g inodeadmgrp

profile: SYSTEM.BROKER.AUTH.IS2
object type: queue
entity: inodeadmgrp
entity type: group
authority: put inq set
- - - - - - - -
profile: SYSTEM.BROKER.AUTH.IS1
object type: queue
entity: inodeadmgrp
entity type: group
authority: put inq set
- - - - - - - -
profile: SYSTEM.BROKER.MB.TOPIC
object type: topic
entity: inodeadmgrp
entity type: group
authority: pub sub
- - - - - - - -
profile: SYSTEM.BROKER.AUTH
object type: queue
entity: inodeadmgrp
entity type: group
authority: put inq set
- - - - - - - -
profile: SYSTEM.BROKER.WEBADMIN.SUBSCRIPTION
object type: queue
entity: inodeadmgrp
entity type: group
authority: get put
- - - - - - - -
profile: self
object type: qmgr
entity: inodeadmgrp
entity type: group
authority: inq connect
- - - - - - - -
profile: @class
object type: queue
entity: inodeadmgrp
entity type: group
authority: none
- - - - - - - -
profile: @class
object type: qmgr
entity: inodeadmgrp
entity type: group
authority: none
- - - - - - - -
profile: @class
object type: topic
entity: inodeadmgrp
entity type: group
authority: none

- - - - - - - -
profile: SYSTEM.BROKER.DEPLOY.QUEUE
object type: queue
entity: inodeadmgrp
entity type: group
authority: put
- - - - - - - -
profile: SYSTEM.BROKER.DC.AUTH
object type: queue
entity: inodeadmgrp
entity type: group
authority: inq set
- - - - - - - -
profile: SYSTEM.BROKER.DEPLOY.REPLY
object type: queue
entity: inodeadmgrp
entity type: group
authority: get put

**Run the following MQ commands to grant full administrative access to all broker resources for the group inodedevgrp.

setmqaut -m YOURQUEUEMANAGER -t qmgr -g inodedevgrp +connect +inq
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.DEPLOY.QUEUE -t queue -g inodedevgrp +put
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.DEPLOY.REPLY -t queue -g inodedevgrp +put +get
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.AUTH -t queue -g inodedevgrp +inq +put +set
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.DC.AUTH -t queue -g inodedevgrp +inq +set
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.WEBADMIN.SUBSCRIPTION -t queue -g inodedevgrp +put +get
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.MB.TOPIC -t topic -g inodedevgrp +sub +pub
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.AUTH.EG -t queue -g inodedevgrp +inq +put +set

**Verify your configurations

dmpmqaut -m YOURQUEUEMANAGER -g inodedevgrp
profile: SYSTEM.BROKER.AUTH.IS2
object type: queue
entity: inodedevgrp
entity type: group
authority: put inq set
- - - - - - - -
profile: SYSTEM.BROKER.AUTH.IS1
object type: queue
entity: inodedevgrp
entity type: group
authority: put inq set
- - - - - - - -
profile: SYSTEM.BROKER.MB.TOPIC
object type: topic

entity: inodedevgrp
entity type: group
authority: pub sub
- - - - - - - -
profile: SYSTEM.BROKER.AUTH
object type: queue
entity: inodedevgrp
entity type: group
authority: put inq set
- - - - - - - -
profile: SYSTEM.BROKER.WEBADMIN.SUBSCRIPTION
object type: queue
entity: inodedevgrp
entity type: group
authority: get put
- - - - - - - -
profile: self
object type: qmgr
entity: inodedevgrp
entity type: group
authority: inq connect
- - - - - - - -
profile: @class
object type: queue
entity: inodedevgrp
entity type: group
authority: none
- - - - - - - -
profile: @class
object type: qmgr
entity: inodedevgrp
entity type: group
authority: none
- - - - - - - -
profile: @class
object type: topic
entity: inodedevgrp
entity type: group
authority: none
- - - - - - - -
profile: SYSTEM.BROKER.DEPLOY.QUEUE
object type: queue
entity: inodedevgrp
entity type: group
authority: put
- - - - - - - -
profile: SYSTEM.BROKER.DC.AUTH
object type: queue
entity: inodedevgrp
entity type: group
authority: inq set
- - - - - - - -
profile: SYSTEM.BROKER.DEPLOY.REPLY
object type: queue

entity: inodedevgrp
entity type: group
authority: get put

**Run the following MQ commands to grant standard access to all broker resources for the group stdusersgrp.
setmqaut -m YOURQUEUEMANAGER -t qmgr -g stdusersgrp +connect +inq
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.DEPLOY.QUEUE -t queue -g stdusersgrp +put
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.DEPLOY.REPLY -t queue -g stdusersgrp +put +get
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.AUTH -t queue -g stdusersgrp +inq
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.DC.AUTH -t queue -g stdusersgrp +inq +set
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.WEBADMIN.SUBSCRIPTION -t queue -g stdusersgrp +put +get
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.MB.TOPIC -t topic -g stdusersgrp +sub +pub
setmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER.AUTH.EG -t queue -g stdusersgrp +inq

**Verify your configuration
dmpmqaut -m YOURQUEUEMANAGER -g stdusersgrp
profile: SYSTEM.BROKER.AUTH.IS2
object type: queue
entity: stdusersgrp
entity type: group
authority: inq
- - - - - - - -
profile: SYSTEM.BROKER.AUTH.IS1
object type: queue
entity: stdusersgrp
entity type: group
authority: inq
- - - - - - - -
profile: SYSTEM.BROKER.MB.TOPIC
object type: topic
entity: stdusersgrp
entity type: group
authority: pub sub
- - - - - - - -
profile: SYSTEM.BROKER.AUTH
object type: queue
entity: stdusersgrp
entity type: group
authority: inq
- - - - - - - -
profile: SYSTEM.BROKER.WEBADMIN.SUBSCRIPTION
object type: queue
entity: stdusersgrp
entity type: group
authority: get put
- - - - - - - -
profile: self
object type: qmgr
entity: stdusersgrp

entity type: group
authority: inq connect
- - - - - - - -
profile: @class
object type: queue
entity: stdusersgrp
entity type: group
authority: none
- - - - - - - -
profile: @class
object type: qmgr
entity: stdusersgrp
entity type: group
authority: none
- - - - - - - -
profile: @class
object type: topic
entity: stdusersgrp
entity type: group
authority: none
- - - - - - - -
profile: SYSTEM.BROKER.DEPLOY.QUEUE
object type: queue
entity: stdusersgrp
entity type: group
authority: put
- - - - - - - -
profile: SYSTEM.BROKER.DC.AUTH
object type: queue
entity: stdusersgrp
entity type: group
authority: inq set
- - - - - - - -
profile: SYSTEM.BROKER.DEPLOY.REPLY
object type: queue
entity: stdusersgrp
entity type: group
authority: get put

If the system queue does not exist use this command to create each one of the below:
runmqsc YOURQUEUEMANAGER
define qlocal (OBJECT_NAME)

Object Type	Object name	Permission Required for Administrator's Role	Permission Required for Normal User's Role
Queue manager	QUEUE_AUTH	Connect, Inquire	Connect, Inquire
Queue	SYSTEM.BROKER.DEPLOY.QUEUE	Put	Put
Queue	SYSTEM.BROKER.DEPLOY.REPLY	Put, Get	Put, Get
Queue	SYSTEM.BROKER.AUTH	Inquire, Put, Set	Inquire
Queue	SYSTEM.BROKER.AUTH.IS1	Inquire, Put, Set	Inquire
Queue	SYSTEM.BROKER.AUTH.IS2	Inquire, Put, Set	Inquire
Queue	SYSTEM.BROKER.DC.AUTH	Inquire, Set	Inquire, Set
Queue	SYSTEM.BROKER.WEBADMIN.SUBSCRIPTION	Put, Get	Put, Get
Topic	SYSTEM.BROKER.MB.TOPIC	Sub, Pub	Sub, Pub

Figure: List of the permissions that can be set

For a complete list of the permissions, read the Setting queue-based permissions on Linux, UNIX, and Windows systems:
https://www.ibm.com/support/knowledgecenter/en/SSMKHH_10.0.0/com.ibm.etools.mft.doc/bp43640_.htm

Refresh the mq security settings
runmqsc YOURQUEUEMANAGER
REFRESH SECURITY (*)

2. (OPTIONAL) Use the dspmqaut command to check administrative security settings for all 3 groups

dspmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER. AUTH -t queue -g inodeadmgrp
or
dmpmqaut -m YOURQUEUEMANAGER -g inodeadmgrp

dspmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER. AUTH -t queue -g inodedevgrp
or
dmpmqaut -m YOURQUEUEMANAGER -g inodedevgrp

dspmqaut -m YOURQUEUEMANAGER -n SYSTEM.BROKER. AUTH -t queue -g stdusersgrp
or
dmpmqaut -m YOURQUEUEMANAGER -g stdusersgrp

PART III: Enabling the Web User Interface and Creating WebUI Accounts

1. Enabling the Webadmin Interface
mqsichangeproperties IIBNODE -b webadmin -o server -n enabled -v true
mqsireportproperties IIBNODE -b webadmin -o server -a

***If the webui is enabled, you should see something like this:*
server=''
uuid='server'
enabled='true'
ldapAuthenticationUri=''
sessionMaxInactiveAgeSecs=''
enableSSL=''

2. (OPTIONAL) Specify the Port Number to Be Used
mqsichangeproperties IIBNODE -b webadmin -o HTTPConnector -n port -v 4423
mqsireportproperties IIBNODE -b webadmin -o HTTPConnector -r
HTTPConnector
uuid='HTTPConnector'
address=''
port='4423'
maxPostSize=''
acceptCount=''
compressableMimeTypes='text/html,text/css,application/javascript,image/gif,image/png,application/json'
compression='on'
connectionLinger=''
connectionTimeout=''
maxHttpHeaderSize=''
maxKeepAliveRequests=''

maxThreads="
minSpareThreads="
noCompressionUserAgents="
restrictedUserAgents="
socketBuffer="
tcpNoDelay="
enableLookups='false'
serverName="
accessLog="
accessLogPattern="
BIP8071I: Successful command completion.

3. Creating web user accounts
Now we will create the web admin users accounts. Remember, that some people rather feed these users id via LDAP servers. Here is where your will be doing that.

Note: You can create as many users as you wish. If you do so, you don't need to create a system user for each one of them. They all will inherent the same system user privileges using -r flag. Here, I only create one user for each group. Also keep in mind that IIB knowledge center repository does not have a documentation on how to set up LDAP with queue-based authentication. If you need assistance doing that, please, contact IBM services for further assistance.

Also, please, read the mqsiwebuseradmin article for more details on the flags that you can use with mqsiwebuseradmin command

Link: https://www.ibm.com/support/knowledgecenter/en/SSMKHH_10.0.0/com.ibm.etools.mft.doc/bn28491_.htm

**Create a user that inherent the system iibadm user privileges created earlier
mqsiwebuseradmin IIBNODE -c -u myAdmin -r iibadm -a password

**Create a user that inherent the system iibdev user privileges created earlier
mqsiwebuseradmin IIBNODE -c -u myDev -r iibdev -a password

**Create a user that inherent the system iibusers user privileges created earlier
mqsiwebuseradmin IIBNODE -c -u stdUser -r iibusers -a password

****Run the following command to list all web users created**
mqsiwebuseradmin IIBNODE -l
BIP2837I: Web user 'myAdmin' is defined as having a role of 'iibadm'. This user will be authenticated against a local password.
BIP2837I: Web user 'myDev' is defined as having a role of 'iibdev'. This user will be authenticated against a local password.
BIP2837I: Web user 'stdUser' is defined as having a role of 'iibusers'. This user will be authenticated against a local password.
BIP8071I: Successful command completion.

PART IV: Cleanup and Logon

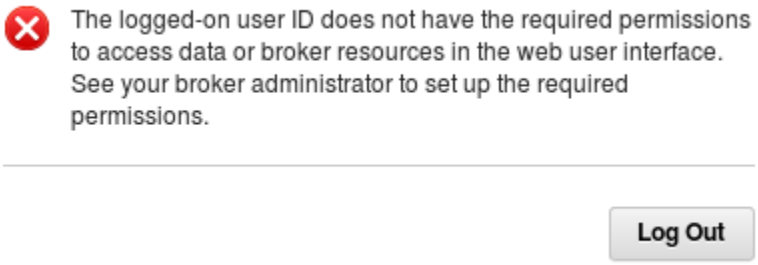
Enabling administrative security
mqsistop IIBNODE
mqsichangebroker IIBNODE -s active
mqsistart IIBNODE

Note: If you experience any issue, remember to go back and make sure you have followed all steps and have not skipped any and that you have refreshed MQ securities after configuring the security permissions.

Keep in mind that IIB knowledge center repository does not have a documentation on how to set up LDAP authentication with IIB. LDAP is a component that is not shipped with IIB and any configuration with such component must be addressed outside of IIB support scope. If you need assistance with that, please, contact IBM services.

PART IV: Troubleshooting

When creating a webadmin user using queue-based authentication, remember that the **-r** must be followed by the *system level user* you have created at the beginning of this document, NOT the system level group that you have used in the **setmqaut** command. For instance, **-r iibadm, -r iibdev, -r iibusers** are all system users NOT groups. If you use a group instead of user, you will bet the below error message.



Error like the one above is always related to queue permission, so make sure you set your queue permissions properly and to the appropriate groups.

***Reload the user security for the broker using mqsireloadsecurity command:

mqsireloadsecurity IIBNODE

Ref: https://www.ibm.com/support/knowledgecenter/en/SSMKHH_10.0.0/com.ibm.etools.mft.doc/an28610_.htm

Another reason for this message to be displayed could be that you did not create the system level queues. In such case creating the system queue and restarting the queue manager should fix the issue.