Unit 12.WebSphere MQ security with OAM

What this unit is about

This unit covers the way WebSphere MQ protects its objects using access control lists (ACLs), and how the OAM uses these ACLs whenever a user attempts to access these objects. The OAM utilities are also explained.

What you should be able to do

After completing this unit, you should be able to:

- Explain the role of object authority manager (OAM) to provide security to WebSphere MQ resources
- Use access control lists (ACLs) to protect WebSphere MQ resources using OAM
- Use some OAM control commands
- Explain how to use the message context information
- Explain the Secure Sockets Layer (SSL) support provided in WebSphere MQ
- Use WebSphere MQ Explorer to control publish/subscribe security

How you will check your progress

Accountability:

- Checkpoint
- Machine exercises

Unit objectives

After completing this unit, you should be able to:

- Explain the role of object authority manager (OAM) to provide security to WebSphere MQ resources
- Use access control lists (ACLs) to protect WebSphere MQ resources using OAM
- Use some OAM control commands
- Explain how to use message context information
- Explain the Secure Sockets Layer (SSL) support provided in WebSphere MQ
- Use WebSphere MQ Explorer to control publish/subscribe security

Figure 12-1. Unit objectives

WM203 / VM2032.0

Notes:

Security overview topic objectives

After completing this topic, you should be able to:

- Explain the role of OAM to provide security to WebSphere MQ resources
- Use ACLs to protect WebSphere MQ resources using OAM

Figure 12-2. Security overview topic objectives WM203 / VM2032.0

Notes:

WebSphere MQ security implementations

- Object Authority Manager (OAM) facility
- Channel security using Secure Sockets Layer (SSL)

Figure 12-3. WebSphere MQ security implementations WM203 / VM2032.0

Notes:

WebSphere MQ security is implemented in two broad areas.

1. **To protect the local WebSphere MQ resources.** Authorization for using MQI calls, commands, and access to objects is provided by the Object Authority Manager (OAM), which by default is enabled. Access to WebSphere MQ entities is controlled through WebSphere MQ user groups and the OAM. You provide a command interface to enable administrations to grant or revoke authorizations as required.
2. **To control access using the network.** Secure Sockets Layer (SSL) protocol provides industry-standard channel security, with protection against eavesdropping, tampering, and impersonation.

WebSphere MQ access control overview

- Granular access control facilities
  - Provided using WebSphere MQ installable services
  - Which user? Which resource? What types of access?
- WebSphere MQ access control at user and group level
  - UNIX use groups only
    - Username must exist, everyone is in nobody
  - Windows uses user IDs and groups
  - System-level user IDs only are supported (no support for DCE principals, TXSeries user IDs, and so forth)
- Alternate user IDs may be specified when suitably authorized
- First level name only is controlled:
  - Alias queues, remote queues
  - Resolved name is not significant

Figure 12-4. WebSphere MQ access control overview WM203 / VM2032.0

Notes:

The primary security component provided by WebSphere MQ is access control. Access control allows WebSphere MQ to control which users are granted which types of access to which WebSphere MQ resources. The resources that might be controlled in this way are the queue manager, queues, and processes.

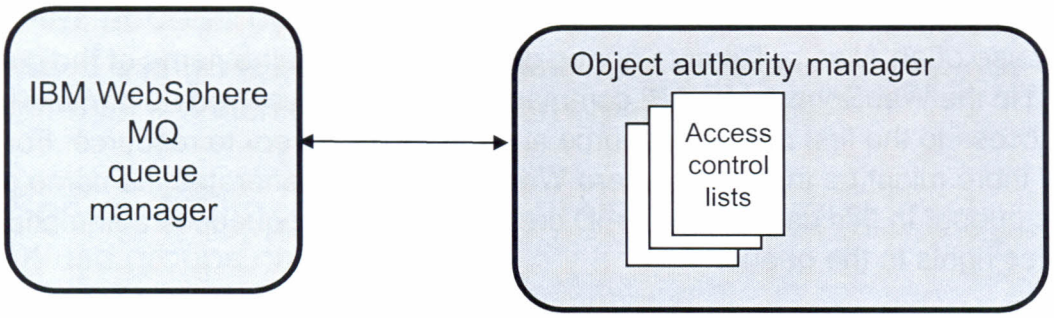
All distributed queue managers provide access control facilities to control which users have access to which WebSphere MQ resources. The distributed queue managers use the Installable Services component of WebSphere MQ - the Authorization Service - to provide access control for WebSphere MQ resources.

WebSphere MQ supplies an Object Authority Manager (OAM) as an authorization service which conforms to the Installable Services interface. The OAM provides a full set of access control facilities for WebSphere MQ including both the access control checking and commands to set, change and inquire on WebSphere MQ access control information. The OAM, like all Installable Services components, is replaceable by any component - user or vendor supplied - that conforms to the Authorization Service interface.

WebSphere MQ captures the user ID associated with an application at MQCONN. This user ID is used for the access control checks. It is possible for suitably authorized users to use an alternate user ID instead of the logged on user ID. When WebSphere MQ checks to see if a user is permitted to access a particular resource, it is the name specified in the WebSphere MQ API command which is used for the check.

For the case of an Alias or Remote queue definition, it is still the name of the queue specified in the WebSphere MQ API command and not the resolved- to name. Thus, a user needs access to the first named resource and not the resolved- to resource. For model queues, there might be instances where **WebSphere MQ** generates the name of the dynamic queue. In this case, the user ID creating a dynamic queue is automatically given full access rights to the queue.

OAM - Installable service



- Documented Interface
  - (User) replaceable
  - Extendable
- Common ACL manager for distributed queue managers
  - UNIX
  - Windows
  - Compaq, OpenVMS, and so forth
  - iSeries

Figure 12-5. OAM - Installable service WM203 / VM2032.0

Notes:

The Object Authority Manager (OAM) component conforms to the WebSphere MQ Installable Services interface.

OAM provides an open, documented interface (documented in the *WebSphere MQ System Administration Guide*).

The OAM is, therefore, user-replaceable, meaning that any component which conforms to the documented interface, may replace the existing OAM.

Further, the OAM (or any component replacing it) may be augmented, having more than one component simultaneously (actually - sequentially) running at the Installable Service Interface.

OAM - Installable service

- Access control lists are WebSphere MQ-specific
  - Not integrated with system-level security
    - Changes to user's OS authority is not recognized until queue manager restart
    - Use `REFRESH SECURITY` command
  - One ACL set per queue manager
    - Not shared between queue managers
- Access control for WebSphere MQ objects
  - Queue manager
  - Queues
  - Processes
  - Namelists
  - Channels
  - Authentication information objects
  - Listeners
  - Services
  - Publish/subscribe topics
- OAM can be disabled
  - Remove entry from `mqs.ini` or Windows registry
  - Not recommended
  - Very difficult to reestablish uniform authority checking

Figure 12-6. OAM - Installable service

WM203 / VM2032.0

Notes:

The implementation of the OAM has a set of associated access control lists.

There is a set of access control lists per queue manager, meaning that the ACLs cannot be (automatically) shared across multiple queue managers.

The OAM provides access control facilities only for WebSphere MQ objects: the queue manager, all queues, processes, namelists, channels, authentication information objects, listeners, and services.

It is possible to disable access control checking by setting the appropriate environment variable or deleting the authorization service stanza in the queue manager configuration file qm.ini (or Registry on Windows systems).

This is not recommended;
If the OAM is disabled, then access permissions which are normally automatically set when objects are created are no longer created. If the OAM is enabled at some later time, these permissions do not exist and have to be created manually.

OAM access control lists

- One authority file per object plus global permissions files
- Each file has one stanza per principal
  - Principal:
  - Authority='bit pattern'
- Windows OAM bypasses authority files for certain classes of principal
  - SYSTEM, local Administrators group, local mqm group
- MQ object permissions
  - 22 permissions
  - Global permissions, for example, connect, altusr, setall, setid
  - Object permissions, for example, browse, put, get ... and allmqi
  - Administration permissions, for example, crt, dsp, chg, dlt ... and alladm
    - Used mostly for PCF commands
  - Plus all and none
- Command `amqoamd` dumps ACLs in readable format

Figure 12-7. OAM access control lists WM203 / VM2032.0

Notes:

Each file contains a set of access control stanzas. There is one stanza per principal for which access is to be controlled, where a principal is either a user ID or a group.

The principals (user IDs, groups, or both) which have access to the appropriate object are listed in this file, along with a bit string (in hex) which represents the access rights associated with that entity.

For each principal which is granted access to an object, there is a permission bit pattern, where each bit corresponds to a particular permission. These permissions are documented in the *WebSphere MQ System Administration*.

Security overview topic summary

Having completing this topic, you should be able to:

- Explain the role of OAM to provide security to WebSphere MQ resources
- Use ACLs to protect WebSphere MQ resources using OAM

Figure 12-8. Security overview topic summary

Notes:

OAM utilities topic objectives

After completing this topic, you should be able to:

- Use some OAM control commands to control the WebSphere MQ security environment

Figure 12-9. OAM utilities topic objectives

WM203 / VM2032.0

Notes:

Security management - setmqaut

- Change authorizations
  - Queue manager
  - Queues
  - Processes
  - Namelists
  - Authorization information (SSL channel security)
  - Channels
  - Listeners
  - Services
  - Publish/subscribe topics
- Principal or group level control
- Granular control of access
  - No generic functions
  - Supports generic profiles

Format:
```
setmqaut -m QMgr -t Objtype -n Profile
  [-p Principal | -g Group] permissions
```

Example:
```
setmqaut -m JUPITER -t queue -n MOON.* -g VOYAGER +browse +get
```


Notes:

There are three control commands which provide control of the security environment for WebSphere MQ, `setmqaut`, `dspmqaut`, `dmpmqaut`.

These programs require a connection to the authorization service and so may only be used when the target queue manager is active and the OAM is enabled. All of the responses to these commands are displayed on the screen. These commands are documented in the *WebSphere MQ System Administration Guide*.

`Setmqaut` is used to set the access that a principal or group has to a particular resource. This command can be used to add or remove privileges.

There are certain principals or groups which are granted automatic access to resources:

- mqm (user/group)
- For Windows:
  - Administrator (user/local group)
  - SYSTEM (user ID)

- The user (or principal group) which creates a resource

All other principals and groups must be granted access to a resource using the utility. **Setmqaut** can use generic profiles. In the example above, the **setmqaut** control command allows members of the group VOYAGER to get and browse messages on the queue whose name commences with the characters MOON that is owned by the queue manager JUPITER. MOON.* is the generic profile.

The PCF equivalent command is Set Authority Record.

Security management - dspmqaut

- Display current authorizations
  - Queue manager
  - Queues
  - Processes
  - Namelists
  - Authorization information (SSL channel security)
  - Channels
  - Listeners
  - Services
- Principal or group level control

Format:

```
dspmqaut -m QMgr -t ObjType -n ObjName
  [-p Principal | -g Group ]
```

Example:

```
dspmqaut -m SATURN -t q -n APPL.Q1 -p mquser
Entity mquser has the following authorizations for object APPL.Q1:
  get
  browse
  put
  ...
```


Figure 12-11. Security management - dspmqaut WM203 / VM2032.0

Notes:

Dspmqaut does support of generic profiles. If a user ID is a member of one or more groups, this command displays the combined authorizations of all the groups.

Only one group or principal can be specified.

On WebSphere MQ for Windows, you can specify a local group.

The PCF equivalent command is *inquire* entity authority.

Security management - dmpmqaut

- Dump current authorizations
  - Queue manager
  - Queues
  - Processes
  - Namelists
  - Authorization information (SSL channel security)
  - Channels
  - Listeners
  - Services
- Principal or group level control

Format:

```
dmpmqaut -m Qmgr -t Objtype -n Profile
         [-p Principal | -g Group]
```

Example:

```
dmpmqaut -m qm1 -n a.b.c -t q -p user1
```

The resulting dump would display:

```
profile: a.b.*
object type: queue
entity: user1
type: principal
authority: get, browse, put ,inq
```


Figure 12-12. Security management - dmpmqaut WM203 / VM2032.0

Notes:

The WebSphere MQ control command enables you to dump the current authorizations associated with a specified profile. The *WebSphere MQ System Administration Guide* provides documentation on this command.

`Dmpmqaut` can be used to dump authority records for a generic profile.

The `-l` parameter allows you to dump only the profile name and the type. The listing is a terse list of all profiles names and types.

Group names must exist and you can only specify one group name on the dmpmqaut command. WebSphere MQ for Windows allows the use of local groups only.

The PCF equivalent command is Inquire Authority Record.

Access control for WebSphere MQ control programs

- Most WebSphere MQ control programs
  For example: `crtmqm`, `strmqm`, `runmqsc`, `setmqaut`
- Have restricted access
  - UNIX restricts users to mqm group
    - Configuration as a part of WebSphere MQ installation
    - Control imposed by the operating system, not OAM
    - MQM group
  - Windows allows
    - mqm group
    - Administrators group
    - System user ID

Figure 12-13. Access control for WebSphere MQ control programs WM203 / VM2032.0

Notes:

Authority checking in the MQI

- MQI calls with security checking
  - MQCONN and MQCONNX
  - MQOPEN
  - MQPUT1 (implicit MQOPEN)
  - MQSUB
  - MQCLOSE (for dynamic queues)
- WebSphere MQ events as audit records
  - Events written to SYSTEM.ADMIN.QMGR.EVENT queue
  - Documented in Monitoring WebSphere MQ manual
- Reason code MQRC_NOT_AUTHORIZED returned if not authorized

Figure 12-14. Authority checking in the MQI WM203 / VM2032.0

Notes:

When attempting to access an alias queue, authority checking occurs at the level of the alias queue, not at the level of the queue to which it resolves. The same is true for a local definition of a remote queue. It is therefore possible to grant no access to a local queue and only grant access to an alias queue which resolves to it.

Limit the ability to define queues to privileged users. Otherwise, normal access control can be bypassed by creating an alias queue.

When an MQSUB call is issued, the queue manager verifies that the user identifier under which the application is running has the appropriate level of authority to subscribe to the topic object.

The MQCLOSE is generally not checked because the close options are typically none.

However, if the close options are set to MQCO_DELETE or MQCO_DELETE_PURGE (only valid for permanent dynamic queues) then, unless the queue was created using the current handle, there is a check to determine if the user is authorized to delete the queue.

Security and distributed queuing

- Put authority option for the receiving end of a message channel
  - Default user identifier is used
  - Context user identifier is used
- Transmission queue
  - Messages destined for a remote queue manager are put on a transmission queue by the local queue manager
    - An application should not normally need to put messages directly on a transmission queue, or need authority to do so
  - Only special system programs should put messages directly on a transmission queue and should have the authority to do so

Figure 12-15. Security and distributed queuing

Notes:

When defining the receiving end of a message channel, there is an option which allows you to specify which user identifier is to be used for checking the authority of the receiving MCA to open a destination queue in order to put a message on it. You can choose one of the following options:

- **Default user identifier**
  The receiving MCAs default user identifier is used. This user identifier may be changed by a security exit, or by setting the MCAUSER parameter in the channel definition at the receiving end of the message channel.
- **Context user identifier**
  The user identifier in the context of the message is used.

Only allow special system programs to put messages directly on a transmission queue.

OAM utilities topic summary

Having completing this topic, you should be able to:

- Use some OAM control commands to control the WebSphere MQ security environment

Figure 12-16. OAM utilities topic summary

WM203 / VM2032.0

Notes:

Message context topic objectives

After completing this topic, you should be able to:

- Explain how to use the message context information to find out about the originator of the message
- Control the use of this facility

Figure 12-17. Message context topic objectives

Notes:

Message context fields

Identity context

- *UserIdentifier*
- *AccountingToken*

Windows	Windows SID (security ID) in compressed format
i5/OS	Job accounting code
UNIX	Numeric user ID in ASCII characters

- *ApplIdentityData*
  - Blank

Origin context

- *PutApplType*
  - MQAT_AIX, MQAT_CICS, and so on
- *PutApplName*
- *PutDate* in YYYYMMDD (GMT)
- PutTime in HHMMSSTH (GMT)
- *ApplOriginData*
  - Blank

Notes:

Message context information allows for the application that retrieves a message to find out about the originator of the message. The retrieving application may want to:

- Check that the sending application has the correct level of authority.
- Keep an audit trail of all the messages it has worked with.

The information is held in two fields: identity context and origin context.

An application can request the queue manager to set the context fields of a message by using the put message option `MQPMO_DEFAULT_CONTEXT` on an MQPUT or MQPUT1 call. This is the default action if no context option is specified.

The visual lists the context fields and provides some examples of what the queue manager sets them to when it generates the information.

Identity context - User that originated the message.

UserIdentifier is 12 characters long. Longer user IDs are generally not permitted. For Windows, the queue manager uses the first 12 characters of the logged-on user name.

For UNIX systems, the queue manager uses the logon name of the application, the effective user identifier of the process if no logon is available, or the user identifier associated with the transaction, if the application is a CICS transaction.

- Accounting token

  The queue manager treats the information in this field as binary information, not character information. When the queue manager generates this information, it sets the first byte to the length of the accounting information. The length of the first byte is in the range zero through 30. The second and subsequent bytes are set to the accounting information based on the environment.

  On Windows, the accounting information is set to a Windows security identifier (SID) in a compressed format.

  On HP NonStop Server and UNIX systems, the accounting information is set to the numeric user identifier, in ASCII characters.

  The last byte is set to the accounting-token type.

- Application data relating to identity.

Origin context

- Type of the application that put the message.
- Name of the application that put the message.

  On Windows systems, the queue manager uses:

  For a CICS application, the CICS transaction name

  For an application that is not a CICS application, the rightmost 28 characters of the fully qualified name of the executable file

  On HP NonStop Server, the rightmost 28 characters, if available otherwise blanks.

  On UNIX systems, it uses the same as for CICS applications just like Windows. For applications that are not CICS applications, the rightmost 14 characters.

- Date when the message was put.

  The format of the date, when it is generated by the queue manager, is YYYYMMDD and the date itself is in GMT.

- Time when the message was put.

  The format of the time, when it is generated by the queue manager, is HHMMSSTH and the time itself is in GMT.

- Application data relating to the origin.

No context

- Requested by put message option
  - MQPMO_NO_CONTEXT
  - Queue manager clears all the context fields, specifically
  - *PutApplType* is set to MQAT_NO_CONTEXT
- To request *default context* or *no context* requires no more authority than that required to put message on queue

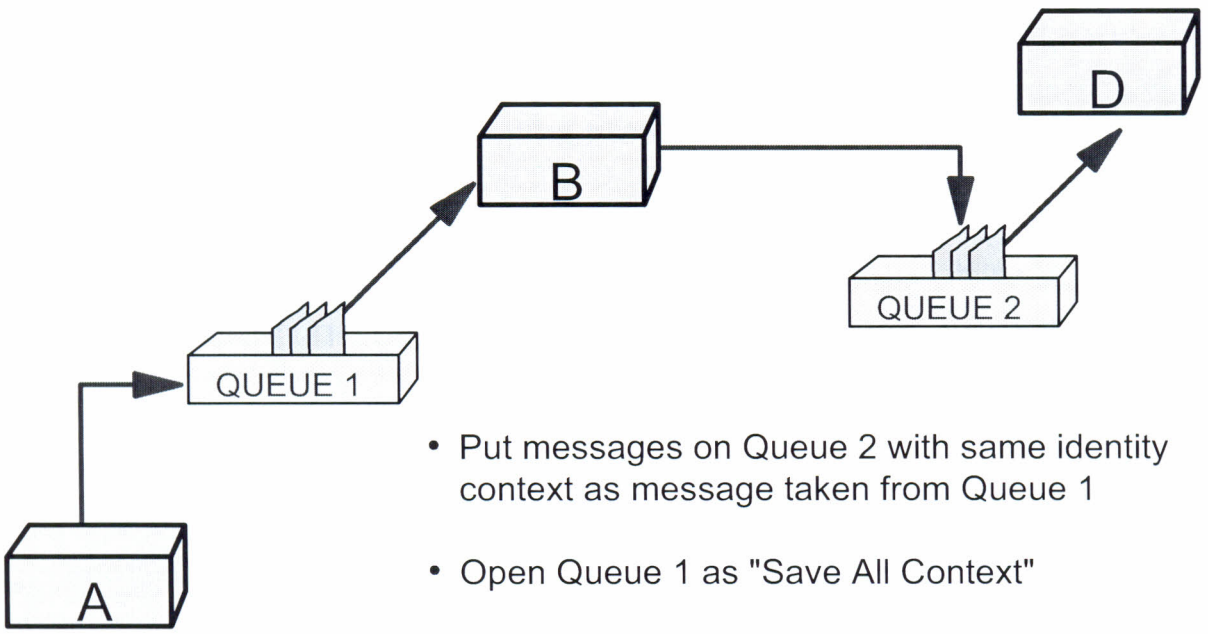
Figure 12-19. No context

Notes:

An application can elect to put a message with no context by specifying the put message option MQPMO_NO_CONTEXT. In this case, the queue manager clears all the context fields. Specifically, it sets the field *PutApplType* to MQAT_NO_CONTEXT so that the receiving application can test for this value. Setting no context can be slightly faster provided the information is not required.

The receiver of a message can test the *PutApplType* field to determine whether the message has no context.

Passing context



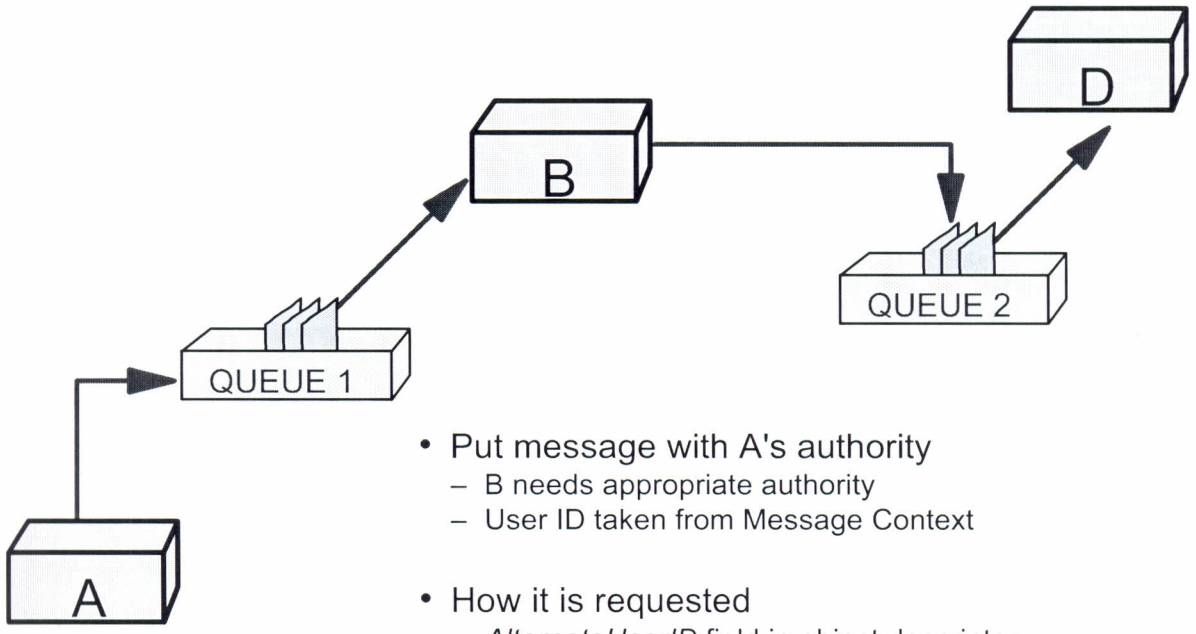
- Put messages on Queue 2 with same identity context as message taken from Queue 1
- Open Queue 1 as "Save All Context"
- Put messages with "Pass Identity Context" or transfer "No Context"

Figure 12-20. Passing context WM203 / VM2032.0

Notes:

Programs generally pass the identity context information from message to message around an application until the data reaches its final destination.

Alternate user authority



- Put message with A's authority
  - B needs appropriate authority
  - User ID taken from Message Context
- How it is requested
  - *AlternateUserID* field in object descriptor
  - Option on `MQOPEN` or `MQPUT1`

Notes:

The alternate user authority option allows an application to open a queue, or any other WebSphere MQ object, by providing the queue manager with a user identifier other than the one it is currently running under. The queue manager uses this alternate user identifier in order to check whether it is authorized to open the queue.