What's new in WebSphere MQ v7.1 Security: A deeper dive

T.Rob Wyatt - t.rob.wyatt@us.ibm.com *or @tdotrob*

Before we get started…

When it comes to security-related information you *always* want to use the latest version! Find the latest version of this deck at http://t-rob.net/links On the same page you can also subscribe to receive update notifications via email when new versions are posted.

Please see the "What's New in WebSphere MQ v7.1 Security" slides as prepared and presented by Morag Hughson at the 2011 WSTC conference in Berlin and available for download at http://t-rob.net/links.

Although there is some overlap, this presentation is intended to cover different ground. Either presentation will stand alone but consider reviewing both for more comprehensive coverage.

Because there is so much material to cover, expect to see more new content with a "What's new in WebSphere MQ v7.1 Security" theme. It may show up as presentations, in articles, video or other formats but all of it will be indexed at t-rob.net where you can subscribe using RSS or via email list.

Thanks to the Global WebSphere Community for hosting!
http://websphereusergroup.org

New security features highlights*

New Feature	Benefits	Details
IP address filtering	Allow or deny connections based on IP address	Blocks IP addresses at the listener and provides allow/deny functionality expressed as per-channel rules
User ID Mapping	Fine grained mapping of connection details to MCAUSER values	Allow or deny connections based on the ID presented in the connection request and map the ID to an MCAUSER
Certificate DN mapping	Finer granularity for matching certificates	Extends SSLPEER functionality to lists of DNs and with expanded regex-type pattern matching and allow/deny capability
User ID blocking	Controls which user IDs can use which channels	After all exits and mapping are completed, a final check is made of the derived MCAUSER against these rules
Authorisation for non-local cluster queues	Simplifies configuration of cluster security	Now possible to define security rules that are enforced against non-local queues for messages that travel through SYSTEM.CLUSTER.XMITQ
Per-channel DLQ settings	Simplifies B2B and other cross-border security	New USEDLQ channel attribute controls which channels will use the DLQ for undeliverable messages.
Additional crypto algorithms	Maintain currency with advancing technology	Adds support for some SHA-2 algorithms and removes some deprecated algorithms now considered to be weak, as reqd by FIPS, Suite-B, others
NSA Suite B support	Addresses suite B requirements	The US National Security Agency (NSA) recommends a set of interoperable cryptographic algorithms in its Suite B standard.
New command: dmpmqcfg	Supported method to back up object configurations and security settings	The fully-supported dmpmqcfg command replaces saveqmgr and amqoamd for v7.1 QMgrs.
New SSLCERTI field in MQCD	New functionality to validate certificate issuer	Provides a method for a security exit to associate a certificate to a particular CA when the KDB contains multiple trusted signers

* This is not by any means a complete list!

What's Changed in WMQ v7.1: http://bit.ly/u1T3E4

Topics covered in this presentation

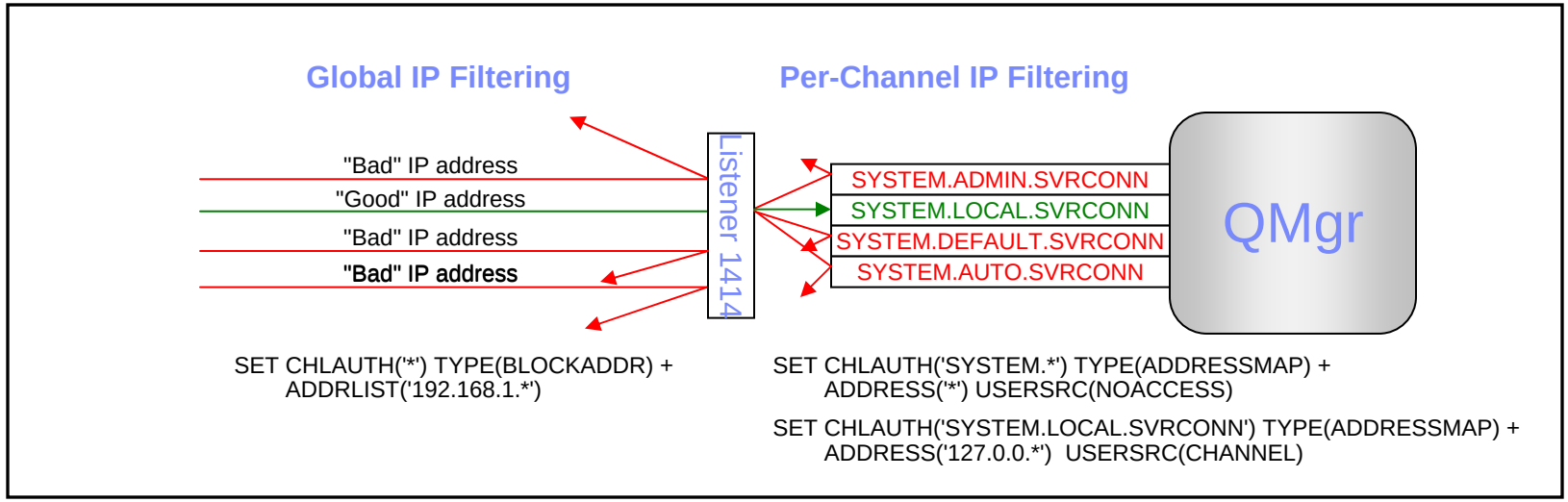
New Feature	*Benefits*	*Details*
IP address filtering	Allow or deny connections based on IP address	Blocks IP addresses at the listener and provides allow/deny functionality expressed as per-channel rules
User ID Mapping	Fine grained mapping of connection details to MCAUSER values	Allow or deny connections based on the ID presented in the connection request and map the ID to an MCAUSER
Certificate DN mapping	Finer granularity for matching certificates	Extends SSLPEER functionality to lists of DNs and with expanded regex-type pattern matching and allow/deny capability
User ID blocking	Controls which user IDs can use which channels	After all exits and mapping are completed, a final check is made of the derived MCAUSER against these rules
Authorisation for non-local cluster queues	Simplifies configuration of cluster security	Now possible to define security rules that are enforced against non-local queues for messages that travel through SYSTEM.CLUSTER.XMITQ

This presentation will focus on the CHLAUTH rules and some detail on how they are used:

- CHLAUTH precedence mapping
- Replacing BlockIP2 channel exit with SSLPEERMAP rules
- Using CHLAUTH QMGRMAP rules to provide granular cluster authorization
- Some notes about migration
- MQ File Transfer Edition interoperability
- WMQ Advanced Message Security interoperability

New feature: IP address filtering

- Prior to v7.1, the ability to validate connection requests based on IP address was only possible using security exits. This functionality is now included natively.
- Provides the ability to filter connection requests based on the IP address of the requestor.
- Two types: Per-channel rules and global blocking rule.
- Global blocking rules occur at the listener before the channel name is known and therefore take precedence over per-channel rules.
- Per-channel rules match from least-specific to most-specific, similar to generic OAM profiles.



IP address filtering guidelines

- Keep in mind: *whitelisting is always better than blacklisting*. A whitelist says "here is an enumerated list of authorized requestors." A blacklist says "out of the practically infinite number of possible requestors, here's a few 'bad' ones." It is impractical to attempt to list all bad addresses.
- CHLAUTH TYPE(BLOCKADDR) is implemented as a blacklist because it is not intended to be the primary means of connection filtering. Use BLOCKADDR to temporarily deal with transient problems such as a runaway client, or to deal with specific issues such as port scanners causing MQ to cut FDC files.
- CHLAUTH TYPE(ADDRESSMAP) rules are hierarchical. With *all other parameters being equal*, the most specific matching profile name takes precedence. This allows you to establish a deny-all policy followed by specific whitelist rules as shown on the previous slide.
- When other parameters are not equal, multiple rules may apply according to a precedence order. This will be covered in detail further on.

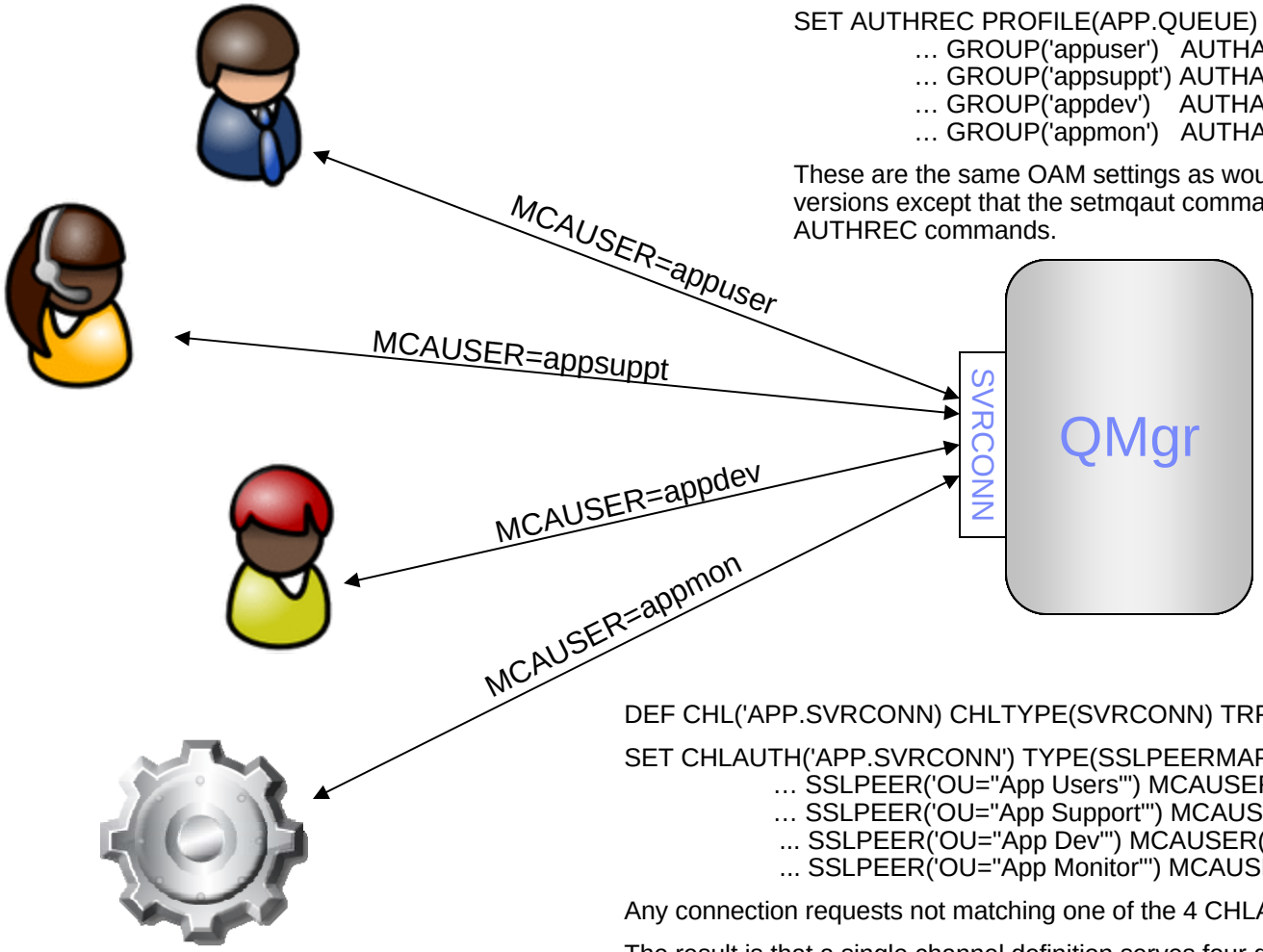
Set CHLAUTH: http://bit.ly/sc83OI
Channel Authentication records: http://bit.ly/veN5C7

Dynamic mapping of MCAUSER

- The channel's MCAUSER determines the ID used for authorization, the same as in previous versions of WebSphere MQ.
- Authorization of remote entities is only as granular as the number of MCAUSER values.
- Prior to v7.1, administrators had a choice of either hard coding the value in the channel definition and defining many channels, or implementing a security exit and configuring the validation criteria outside of WebSphere MQ.
- New in WebSphere MQ v7.1 is the ability to configure dynamic mapping of the MCAUSER based on a variety of validation criteria. This allows the use of fewer channels to support the same or even finer granularity of authorization than was available in prior versions, and with all configuration details managed natively using standard WebSphere MQ tools.

Set CHLAUTH: http://bit.ly/sc83OI
Channel Authentication records: http://bit.ly/veN5C7

Example of dynamic MCAUSER mapping




```
SET AUTHREC PROFILE(APP.QUEUE) OBJTYPE(QUEUE) +
        … GROUP('appuser')   AUTHADD(INQ, DSP, BROWSE, PUT, GET)
        … GROUP('appsuppt') AUTHADD(INQ, DSP, BROWSE)
        … GROUP('appdev')    AUTHADD(INQ, DSP)
        … GROUP('appmon')   AUTHADD(INQ)
```

These are the same OAM settings as would have been required in prior versions except that the setmqaut commands have been migrated to v7.1 SET AUTHREC commands.

```
DEF CHL('APP.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP) MCAUSER('nobody')

SET CHLAUTH('APP.SVRCONN') TYPE(SSLPEERMAP) +
        … SSLPEER('OU="App Users"') MCAUSER('appuser')
        … SSLPEER('OU="App Support"') MCAUSER('appsuppt')
        ... SSLPEER('OU="App Dev"') MCAUSER('appdev')
        ... SSLPEER('OU="App Monitor"') MCAUSER('appmon')
```

Any connection requests not matching one of the 4 CHLAUTH records are refused.

The result is that a single channel definition serves four different security roles for the same application. The DEF CHL, SET CHLAUTH and SET AUTHREC definitions are all managed using standard MQ admin tools, possibly all in the same MQSC script.

SSLPEER Mapping

- The channel's SSLPEER attribute filters connections based on the certificate Distinguished Name of the connection requestor. The SSLPEER was limited to a single value (although wild cards are supported) and does not associate that value with a particular MCAUSER other than by hard-coding the MCAUSER in the channel definition.
- In v7.1, one or more CHLAUTH SSLPEERMAP rules may be defined. SSLPEERMAP rules can either map certificate distinguished names to MCAUSER values to allow access, or specify certificate distinguished names for which access is to be denied.
- SSLPEERMAP rules are hierarchical so it is possible to set a blanket rule to establish a "deny all" policy, override that with a generic access rule and then override that with a specific deny access rule. For example:

```
SET CHLAUTH(*) TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
SET CHLAUTH(*) TYPE(SSLPEERMAP) SSLPEER('OU=ADMIN,O=IBM') USERSRC(CHANNEL)
SET CHLAUTH(*) TYPE(SSLPEERMAP) SSLPEER('CN="T.ROB",OU=ADMIN,O=IBM') USERSRC(NOACCESS)
```

Since these rules limit connections to administrators, we are accepting the ID presented in the connection request.
A more comprehensive rule set would probably use much more specific channel names after the first rule.

User ID blocking

- The CHLAUTH BLOCKUSER rules take effect after all other rules and exits have been processed and the final value for MCAUSER has been determined.
- A special value *MQADMIN represents administrative users as defined for the local platform. This makes locking down admin access much easier – just block *MQADMIN!
- A blank user ID will resolve to the ID of the MCA which then matches *MQADMIN.
- Rules are hierarchical and the most specific one matches.
- This is a blacklist-only setting. However, it is possible to implement a limited deny/allow policy by altering the list of blocked names at different levels. For example:

```
SET CHLAUTH(*) TYPE(BLOCKUSER) USERLIST('nobody, *MQADMIN')
SET CHLAUTH(SYSTEM.ADMIN.*) TYPE(BLOCKUSER) USERLIST('nobody')
```

The first rule blocks administrative users and the MCAUSER 'nobody' (which prevents someone from creating a user ID 'nobody' and putting it into an authorized group.

The second rule provides a reduced blacklist for SYSTEM.ADMIN channels that allows administrators to use these. It is assumed here that some other CHLAUTH rule such as an SSLPEERMAP has validated the administrator's connection or than an exit has done so..

Set CHLAUTH: http://bit.ly/sc83OI
Channel Authentication records: http://bit.ly/veN5C7

Access control for non-local cluster queues

- Granting access to put messages directly to the cluster transmit queue implicitly grants access to address messages to all queues of adjacent cluster queue managers. This led to the practice of defining local QALIAS or QREMOTE objects to improve security on the sending queue manager.
- It is now possible to grant access to queues accessed through the cluster transmit queue without granting access to S.C.T.Q itself.
- Queue name must resolve locally to SYSTEM.CLUSTER.TRANSMIT.QUEUE.
- Does not replace authorization at the receiving QMgr! It is still a good idea to have a low-privileged MCAUSER on the CLUSRCVR channel.

Remote object authorisation: http://bit.ly/v3vaMY

CHLAUTH precedence rules

CHLAUTH Rule Types		
Precedence	**Type**	**Where**
1	BLOCKADDR	Listener
2	BLOCKUSER	Channel
3	*MAP	Channel

Life cycle of a connection request

1. The request first hits the listener where any BLOCKADDR rules are enforced.
2. The request is passed to the queue manager where any mapping rules and exits are applied.
3. After all other processing, the BLOCKUSER rules are applied based on the value of MCAUSER that has been asserted, mapped by a CHLAUTH rule or set by an exit.
4. If the connection request has not been denied in one of the previous steps, an attempt to start the channel is made.

Channel Authentication records: http://bit.ly/veN5C7

CHLAUTH MAP precedence rules

CHLAUTH Rule Types		
Precedence	**Type**	**Where**
1	BLOCKADDR	Listener
2	BLOCKUSER	Channel
3	*MAP	Channel

CHLAUTH Map Types		
Order	**Map Type**	**Identity Mechanism**
0		Channel name
1	SSLPEERMAP	Certificate Distinguished Name
2=	USERMAP	Client asserted ID
2=	QMGRMAP	Queue Manager name
4	ADDRESSMAP	IP address

Where multiple map rules match the same element, the most specific one takes precedence. For example, the following are in order of increasing precedence:

```
PROFILE('*')
PROFILE('SYSTEM.*')
PROFILE('SYSTEM.ADMIN.SVRCONN')
```

Channel Authentication records: http://bit.ly/veN5C7

CHLAUTH precedence rules

CHLAUTH Map Types		
Order	Map Type	Identity Mechanism
0		Channel name
1	SSLPEERMAP	Certificate Distinguished Name
2=	USERMAP	Client asserted ID
2=	QMGRMAP	Queue Manager name
4	ADDRESSMAP	IP address

Distinguished Name Elements		
Order	**DN element**	**Name**
1	SERIALNUMBER=	Serial number
2	MAIL=	Email address
3	E=	Email address (Deprecated)
4	UID=, USERID=	User Identifier
5	CN=	Common Name
6	T=	Title
7	OU=	Organizational Unit
8	DC=	Domain Component
9	O=	Organization
10	STREET=	Street / First line of address
11	L=	Locality
12	ST=, SP=, S=	State/Province
13	PC=	Postal code / ZIP code
14	C=	Country
15	UNSTRUCTUREDNAME=	Host name
16	UNSTRUCTUREDADDRESS=	IP address
17	DNQ=	Distinguished name qualifier

Moving BlockIP2 function into channel config

- The most common use of BlockIP2 is to map certificate Distinguished Names or IP addresses to MCAUSER values
- BlockIP2 configuration is managed in text files on each WMQ server
- CHLAUTH rules can replace most instances of BlockIP2
- In addition, CHLAUTH rules consolidate the configuration into the MQSC script, remote management tool (i.e. WMQ Explorer) or central management tool (i.e. Tivoli)

Set CHLAUTH: http://bit.ly/sc83OI
Channel Authentication records: http://bit.ly/veN5C7

Moving BlockIP2 function into channel config

```
# BlockSpec.txt

# Allow the certificates from the administrator's OU
SSL=CN=*,OU=ADMIN,O=IBM;MCA=*; ok userid

# Allow the certificates from the payroll app
SSL=CN=Payroll,OU=PAYROLL,O=IBM;MCA=payroll;

# Block everything else
SSL=CN=*;BLOCK;        blocked user
```

A BlockSpec.txt file that maps certificate DNs to MCAUSER IDs and its plug-and-play CHLAUTH replacement

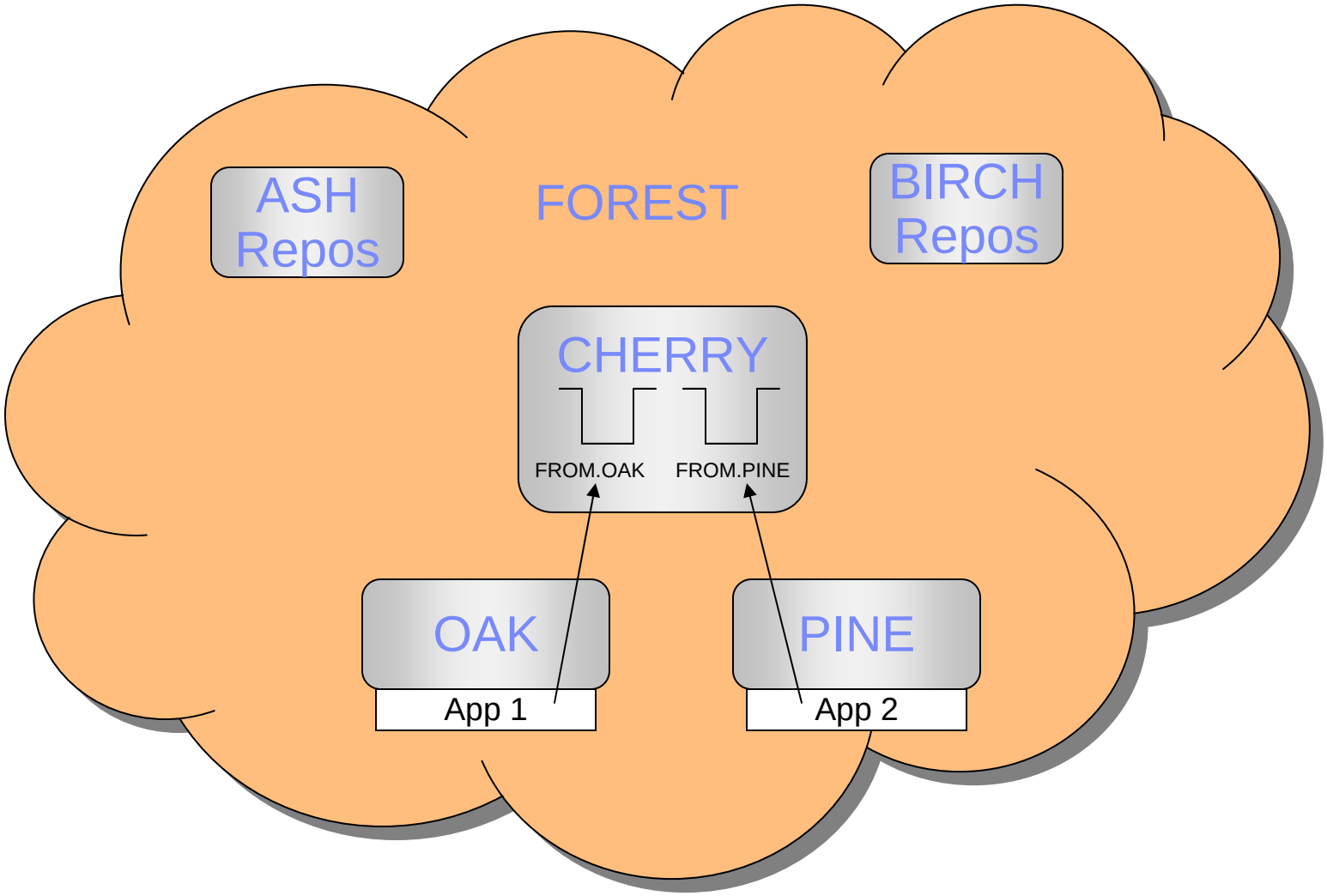
```
* Allow the certificates from the administrator's OU
SET CHLAUTH('*') TYPE(SSLPEERMAP) SSLPEER('CN=*,OU=ADMIN,O=IBM') +
    USERSRC(CHANNEL) ACTION(ADD)

* Allow the certificates from the payroll app
SET CHLAUTH('*') TYPE(SSLPEERMAP) SSLPEER('CN=Payroll,OU=PAYROLL,O=IBM') +
    USERSRC(MAP) MCAUSER('payroll') ACTION(ADD)

* Block everything else
SET CHLAUTH('*') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS) ACTION(ADD)
```

Set CHLAUTH: http://bit.ly/sc83OI
Channel Authentication records: http://bit.ly/veN5C7

QMgr name mapping use case: Granular cluster security

- In prior versions, the MCAUSER of a CLUSRCVR had one value for all remote nodes. The result was that any QMgr in the cluster had access to all queues served by that CLUSRCVR.
- Alternatives included multiple overlapping clusters or a channel auto-definition exit. Creating multiple clusters was burdensome on administrators and added complexity. The CHAD exit option was simpler but was not available from IBM.
- Because of these issues, many shops either did without granular cluster security or avoided clusters altogether.
- The availability of CHLAUTH mapping allows per-node granularity in securing the cluster. Each adjacent cluster node can be identified by a distinct MCAUSER value

Demonstration Cluster layout



Configuration on CHERRY QMgr

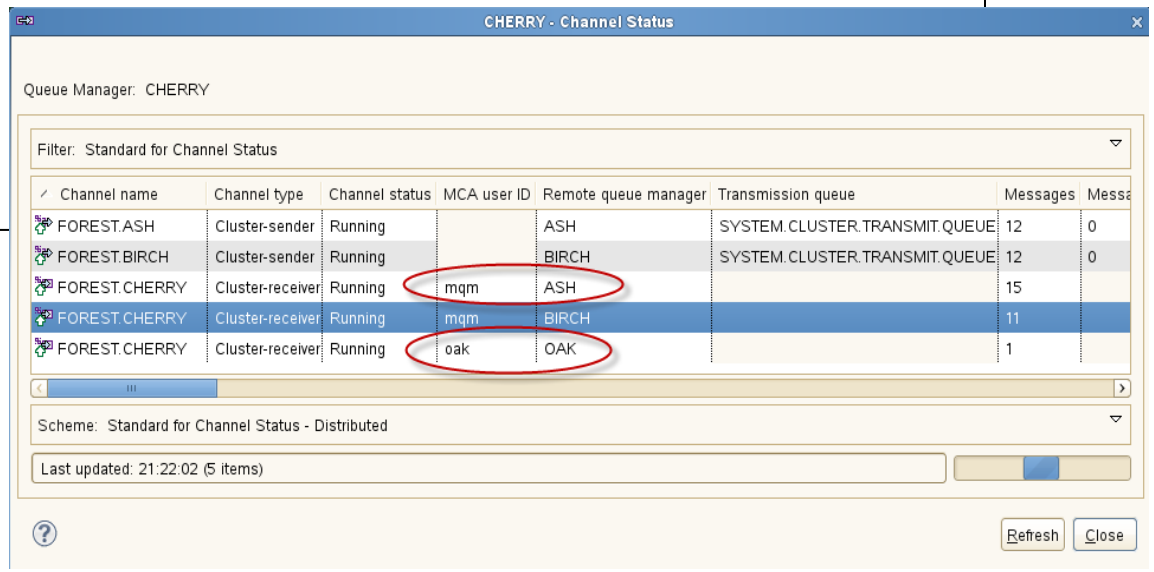
```
* MQSC Script

* Cluster queues
DEF QL(FROM.OAK) CLUSTER(FOREST) +
    DEFPSIST(YES) DEFBIND(NOTFIXED) +
    REPLACE
DEF QL(FROM.OAK) CLUSTER(FOREST) +
    DEFPSIST(YES) DEFBIND(NOTFIXED) +
    REPLACE

* Authorizations
SET AUTHREC OBJTYPE(QMGR) GROUP('forest') AUTHADD(CONNECT, SETALL)
SET AUTHREC PROFILE(CHERRY.DEAD.LETTER.QUEUE) OBJTYPE(QUEUE) +
    GROUP('forest') AUTHADD(PUT, SETALL)

SET AUTHREC PROFILE(FROM.OAK)  OBJTYPE(QUEUE) GROUP('oak')   AUTHADD(PUT, SETALL)
SET AUTHREC PROFILE(FROM.PINE) OBJTYPE(QUEUE) GROUP ('pine') AUTHADD(PUT, SETALL)

* Map remote QMgr name as MCAUSER
SET CHLAUTH('FOREST.CHERRY') TYPE(QMGRMAP) QMNAME('OAK')  USERSRC(MAP) MCAUSER('oak')
SET CHLAUTH('FOREST.CHERRY') TYPE(QMGRMAP) QMNAME('PINE') USERSRC(MAP) MCAUSER('pine')
```

IDs and groups defined for MCAUSER and OAM on CHERRY server

User ID	Groups
oak	forest, oak
pine	forest, pine

Testing the security

```
mqm@SLES11SP1-64:> echo 'Success!!!!!' | q -ap -m OAK -oCHERRY/FROM.OAK
MQSeries Q Program by Paul Clarke [ V5.0.0 Build:Jul 17 2008 ]
Connecting ...connected to 'OAK'.
>>mqm@SLES11SP1-64:q -m CHERRY -iFROM.OAK
MQSeries Q Program by Paul Clarke [ V5.0.0 Build:Jul 17 2008 ]
Connecting ...connected to 'CHERRY'.
Success!!!!!
No more messages.
mqm@SLES11SP1-64:>
```

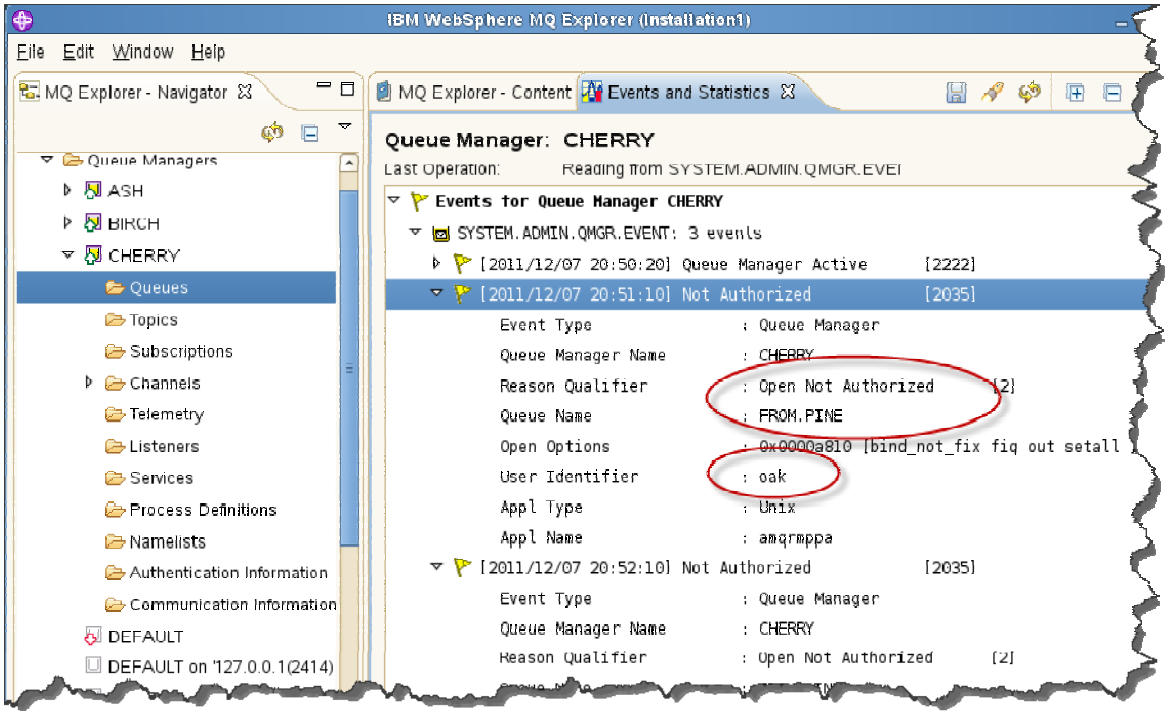


- Connect to OAK and put a message to the FROM.OAK queue on CHERRY
- Connect to CHERRY and read back the message just put there
- Note that the channels from the repositories are running as mqm which is the default.
- The channel from OAK is running as MCAUSER('oak') as dictated by the CHLAUTH rule
- The message was successfully put to the FROM.OAK queue, as expected.
- But how do we know that a message from the "wrong" QMgr will fail? See the next slide.

Testing the security

```
mqm@SLES11SP1-64:> echo 'FAIL!!!!!'  |  q -ap -m OAK -oCHERRY/FROM.PINE
MQSeries Q Program by Paul Clarke [ V5.0.0 Build:Jul 17 2008 ]
Connecting ...connected to 'OAK'.
>>mqm@SLES11SP1-64:>
```

- Connect to OAK and put a message to FROM.PINE on CHERRY
- The CHLAUTH rule maps the channel MCAUSER to 'oak'
- The channel is not authorized to put messages on the FROM.PINE queue We can verify this by checking the event messages on CHERRY
- The CLUSSDR from OAK to CHERRY will go to retry if the messages are persistent



Download the Q program: http://ibm.co/SupptPacMA01

Other uses for granular cluster security

```
* MQSC Script


* Authorizations
SET AUTHREC OBJTYPE(QMGR) GROUP('repos') AUTHADD(CONNECT, SETALL)
SET AUTHREC PROFILE(SYSTEM.CLUSTER.COMMAND.QUEUE) +
    OBJTYPE(QUEUE) GROUP('repos')   AUTHADD(PUT, SETALL)

* Map remote QMgr name as MCAUSER
SET CHLAUTH('FOREST.CHERRY') TYPE(QMGRMAP) QMNAME('ASH')   USERSRC(MAP) MCAUSER('repos')
SET CHLAUTH('FOREST.CHERRY') TYPE(QMGRMAP) QMNAME('BIRCH') USERSRC(MAP) MCAUSER('repos')
```

There are many good reasons to host full repository queue managers on dedicated servers. Add to these, as of WMQ v7.1, the ability to restrict access to SYSTEM.CLUSTER.COMMAND.QUEUE to only those channel instances that come from legitimate repository queue managers. A partial repository queue manager should never accept cluster commands from a queue manager other than one of the two legitimate full repositories.

Note that the full repository queue managers are the exception because they must allow access to S.C.C.Q from all cluster member queue managers. This is how the members advertise their object state changes to the cluster. Do NOT apply the rules above on the full repsoitories.

Migrating from v6.0/v7.0 to v7.1

- SSLPEER OU order has been reversed/corrected. In prior versions of WebSphere MQ there was a discrepancy on distributed between the order you would enter OU elements when specifying SSLPEER versus the order that they were evaluated in by the channel. This required the administrator to reverse the elements when defining SSLPEER. The discrepancy has been corrected so that SSLPEER elements can now be entered in their natural order. Unfortunately, this breaks channels when migrating.
  SSLPEER and SSLCERTI changes: http://bit.ly/tEjItP
- Channels may break if SSLFIPS enabled and a deprecated FIPS algorithm was used. Some ciphersuites that were previously approved for FIPS are now considered weak and no longer allowed to run when SSLFIPS(YES) is set at the queue manager. If these ciphersuites are specified on channels on a FIPS-enabled queue manager, they will fail to start after upgrading to v7.1.
  Some FIPS 140-2 compliant channels do not start: http://bit.ly/w4jVlc
- Enable CHLAUTH at the QMgr. To maintain backward compatibility when queue managers are migrated, the queue manager's CHLAUTH attribute is set to DISABLED. If you wish to use CHLAUTH rules on a migrated queue manager, CHLAUTH must be enabled.
  Channel authentication: http://bit.ly/t3q2oa

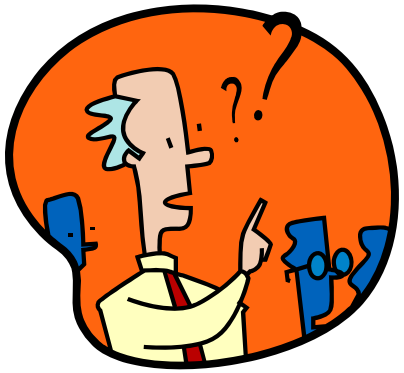
WebSphere MQ FTE / V7.1 Security Interop

- WMQ File Transfer Edition is completely compatible with WMQ v7.1 security.
- WMQ FTE relies on the value of the MQMD.UserID to be trustworthy. This implies that the network has been secured against unauthorized connections from both clients and QMgrs.
- Validation process:
  - For each QMgr, verify that all SVRCONN channels with administrative access are strongly authenticated, preferably using certificates and fully-qualified SSLPEERMAPs.
  - For each QMgr, verify that all other SVRCONN channels with access to FTE queues have been authenticated and a low-privileged MCAUSER has been assigned. Use CHLAUTH rules with *MQADMIN to block administrative users on these channels.
  - For RQSTR, RCVR and CLUSRCVR channels, verify that they authenticate the remote QMgr and map a low-privileged MCAUSER.
  - Repeat these steps on each adjacent remote QMgr.
- If the existing FTE network uses certificates and an exit, these can continue as-is after migration or migrate the exit to CHLAUTH rules and enable CHLAUTH at the QMgr.

WebSphere MQ AMS / V7.1 Security Interop

- WMQ Advanced Message Security is not currently compatible with WMQ v7.1, however this is a top priority and support is expected soon.
- Remember that WMQ provides security at the connection and AMS provides security at the message. This means that AMS *extends* the functionality of base WMQ, does not replace it. In other words, you can have WMQ native security without AMS but you cannot have AMS security without native WMQ security enabled.
- The bare minimum WMQ base security required to use AMS effectively is to lock down administrative access: use CHLAUTH *MQADMIN rules to block admin access from regular channels and strongly authenticate any dedicated admin channels. This prevents non-admins from changing AMS configuration and policy settings.
- The additional functionality in WMQ v7.1 streamlines the administrative tasks required to properly secure the network and consolidates object definitions, authentication rules and authorization policies into a single script or tool.

Questions?



Notices

Information provided has been developed as a collection of the experiences of technical services professionals over a wide variety of customer and internal IBM environments, and may be limited in application to those specific hardware and software products and levels

The information contained in this document has not been submitted to any formal IBM test. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk, and in some environments may not achieve all the benefits described.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of this publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may not offer the products, services, or feature discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials of this IBM product and use of those Web sites is at your own risk.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice and represent goals and objectives only.

Any performance data contained in this document was determined in a controlled environment. Therefore the results obtained in other operating environments may vary significantly. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee that these measurements will be the same on generally available systems. Some measurements quoted in the document may have been estimated through extrapolation. Actual results may vary. Users of this presentation should verify the applicable for their specific environment.

Version History

20111208 – TRW – First release.
20111220 – TRW – Changed curly quotes to straight quotes.