WebSphere MQ Base Hardening

T.Rob Wyatt

t.rob@ioptconsulting.com

Change is the only constant

This presentation reflects…

- **My current opinions regarding WMQ security**
- **The product itself continues to evolve (even in PTFs)**
- **Attacks only get better with time**
- **This version of the presentation is based on WebSphere MQ v7.1 & v7.5**
- **This content will be revised over time so please be sure to check for the latest version at https://t-rob.net/links**
- **Your thoughts and ideas are welcome**

WebSphere MQ Security Presentation Series

- This presentation is part of a series authored by T.Rob Wyatt. The introductory decks are available through https://t-rob.net.
- For the most current version of this deck, please see https://t-rob.net/links or contact the author presentations@t-rob.net.
- Ask me about on-site and remote training tailored for your organization.

Why base hardening?

- **By now we all know that…**
  - Queue managers are designed to *do* something on the arrival of a message.
  - That includes the ability to execute OS commands.
  - Anyone who can gain administrative access can therefore execute arbitrary OS commands through the queue manager, using the WMQ administrative ID.
  - Therefore that admin access had better be authenticated!

- **This session will therefore discuss the architecture & security model first, then move into some of the new v7.1+ base hardening capabilities.**

At IMPACT we had 25 slots for WebSphere Messaging content. Here we have 70! But that means we will need to get a feel for the level of presentation that best suits this audience – and I didn't think the intro deck was appropriate.

Also, the sessions I'm presenting have fewer slides because we always tend to run out of time for Q&A.

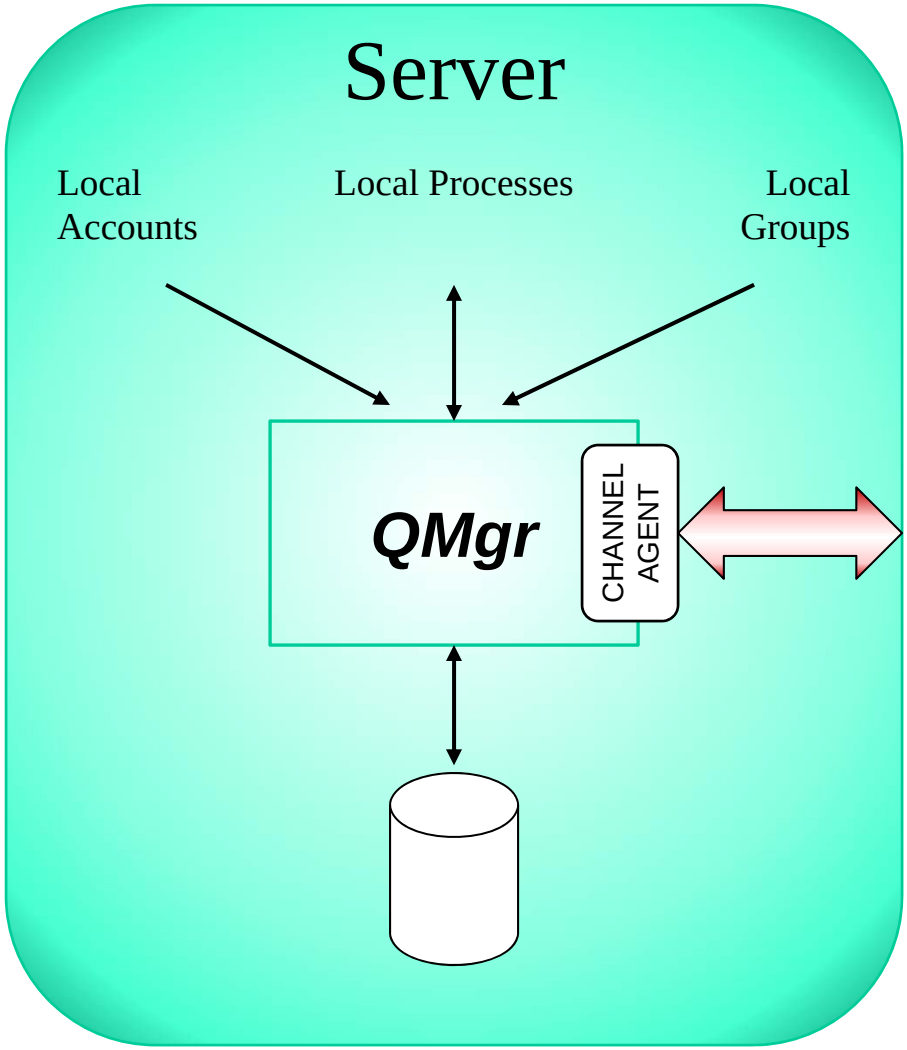
If you wanted the Intro deck, please see me after the session!
If you wanted more written content & less Q&A, please let me know!

Agenda

- **WebSphere MQ Architecture**
- Deep dive into CHLAUTH rules
- CHLAUTH Precedence
- CHLAUTH use case: Granular cluster security
- Managing certificate expiry
- Other new GSKit 8 features
- Additional Resources

WebSphere MQ Architecture

- Designed as a LOCAL transport
- Processes authenticated by OS
- Channels use local agent on behalf of remote entity
- Channels ALWAYS run with administrative authority
- Authentication based on an alternate ID nominated by agent
- ID and group must exist locally



Where are remote connections authenticated?

- **The WMQ client code uses the ID of the client process**
- **But it is trivial for the client to present an administrative ID!**
- **If the ID presented is trusted, the remote partner can administer the queue manager (including remote code execution)**
- **MQ administrator is responsible for configuring meaningful authentication and tying it to authorization**
  - Hard-code the channel's MCAUSER
  - Use CHLAUTH rules or an exit to map an ID to the MCAUSER

The authentication must always occur at the queue manager.

The Object Authority Manager

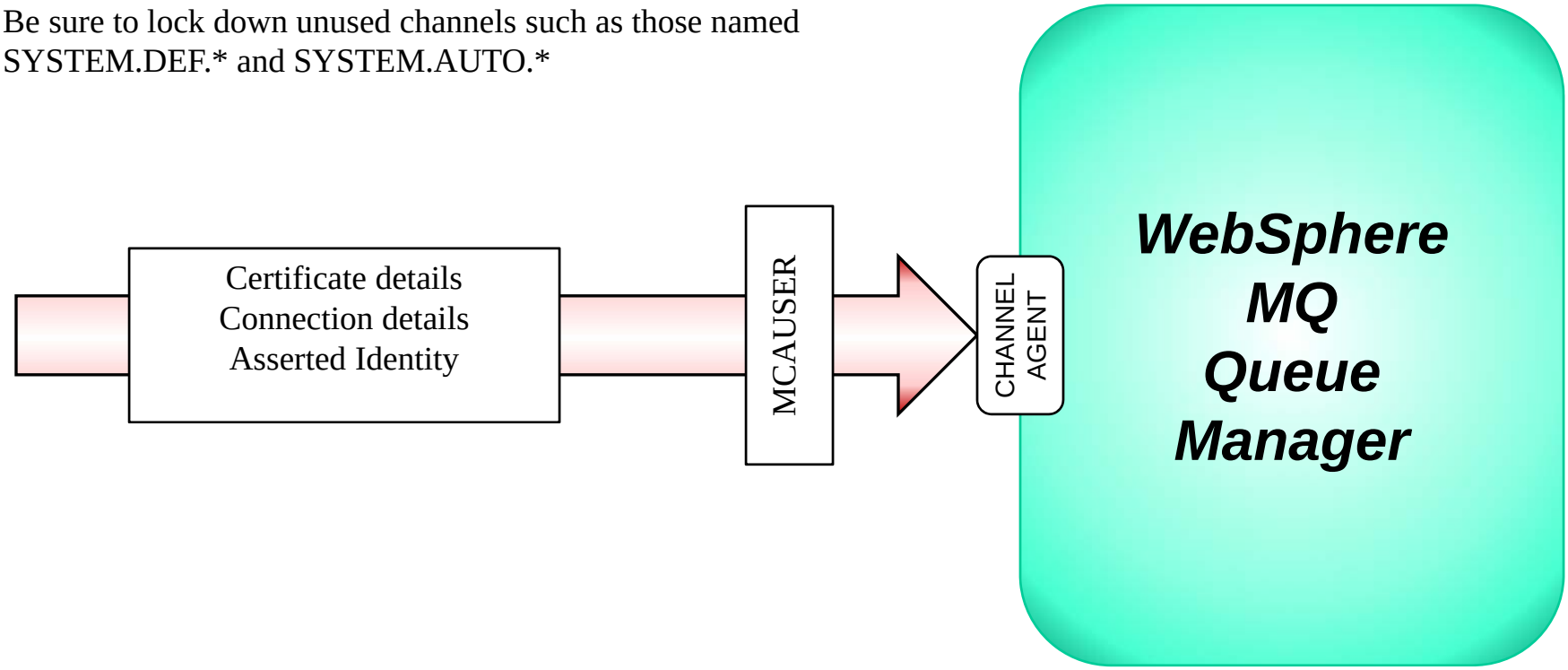
- **The OAM is a pluggable service that makes run-time authorization decisions based on access control lists**
- **Typical UNIX-type authorization supports separation of duties:**
  - Resource admin grants groups rights to resources such as queue managers and queues
  - Account admin enrolls user IDs into groups
- **Supports fine-grained authorizations against WMQ resources**
- **Assumes that some meaningful authentication has been performed!**
  - The OS authenticates IDs of local processes
  - The WMQ admin must provide for authentication of remote connection requests

WMQ remote authentication

Mapping of the MCAUSER is based on any of several values carried on the connection request.

Exits or CHLAUTH rules map the connection attributes to the appropriate MCAUSER value.

Be sure to lock down unused channels such as those named SYSTEM.DEF.* and SYSTEM.AUTO.*



Authentication and mapping tools

- **Hard-code the MCAUSER**
- **The channel's SSLPEER attribute can filter connections based on certificate Distinguished Name**
- **In V7.0 and prior, a channel security exit such as BlockIP2 was required for any more sophisticated authentication**
- **As of V7.1, CHLAUTH rules can evaluate connection details and map these to MCAUSER values or reject the connection.**
- **CHLAUTH also provides**
  - IP blacklisting at the listener
  - User ID blacklisting at the queue manager
  - Generic role represents any administrative connection

Agenda

- WebSphere MQ Architecture
- Deep dive into CHLAUTH rules
- CHLAUTH precedence
- CHLAUTH use case: Granular cluster security
- Managing certificate expiry
- Other new GSKit 8 features
- Additional Resources

Blocking IP Addresses

Blocking implements a blacklist against specific named addresses

Blocking can occur at the listener or at the channel

- Blocking at the listener is per-QMgr
- Blocking at the channel is per-channel

```
# Block at listener for whole QMgr
SET CHLAUTH('*') TYPE(BLOCKADDR) +
    ADDRLIST(generic-ip-address)

# Block at channel for one or more channels
SET CHLAUTH(generic-channel-name) TYPE(ADDRESSMAP) +
    ADDRLIST(generic-ip-address) USERSRC(NOACCESS)
```

Blocking at the listener stops the connection before it reaches the queue manager. Helpful for network pingers, runaway clients, etc.

Filtering IP Addresses

Use CHLAUTH ADDRESSMAP with USERSRC(CHANNEL) to implement a whitelist of addresses.

```
# Filter at channel for one or more channels
SET CHLAUTH(generic-channel-name) TYPE(ADDRESSMAP) +
    ADDRLIST(generic-ip-address) USERSRC(CHANNEL)
```

Example:

```
# Use a blocking rule to set a deny-all policy
SET CHLAUTH(*) +
    TYPE(ADDRESSMAP) ADDRLIST(*) USERSRC(NOACCESS)
# Override with a filter rule to allow the Tivoli
server
SET CHLAUTH(TIVOLI.SVRCONN) TYPE(ADDRESSMAP) +
    ADDRLIST(1.2.3.4) USERSRC(CHANNEL)
```

See the section on Precedence for more details of how multiple rules combine.

Mapping Is One Way to Bind an Identity to a Channel

- **Sets the MCAUSER of a channel based on the …**
  - Certificate distinguished name
  - IP address of the requestor (as determined from the IP socket)
  - ID of the requestor. This can be either of:
    - Client user ID provided in the connection request
    - Name of the remote QMgr
- **Overrides MCAUSER value specified in the channel definition**
- **OAM uses the resolved MCAUSER for authorization checks**
- **When mapping on Windows, specify the ID as `userid@domain` or `userid@domain` so that it is not ambiguous**

Setting MCAUSER by hard-coding, by using exits or by CHLAUTH mapping prevents ID spoofing!

Mapping Certificate Distinguished Names

```
# Map one or more DN to MCAUSER
SET CHLAUTH(generic-channel-name) +
    TYPE(SSLPEERMAP) SSLPEER(generic-ssl-peer-name) +
    USRSRC(MAP) MCAUSER('user-id')
```

- Specify one or more fields of the Distinguished Name in the SSLPEER attribute
- The SSLPEER specification can be fully qualified to match a single DN, with one rule per certificate
- Create a generic SSLPEER using a wildcard or by omitting fields
  - Match strings may contain wildcard (asterisk '*') at the beginning, end, or both places
  - Omitted fields are considered a match

Mapping IP Addresses

```
# Map one or more DN to MCAUSER
SET CHLAUTH(generic-channel-name) +

    TYPE(ADDRESSMAP) ADDRESS('generic-ip-address') +

    USRSRC(MAP) MCAUSER('user-id')
```

- Specify a fully-qualified or generic IP address
- The IP address *as seen by the QMgr*
- Most useful for SVRCONN channels with many instances

IP mapping is useful but is not authentication.
Prefer stronger authentication methods where possible.

Mapping Client User ID

```
# Map a single client ID to MCAUSER
SET CHLAUTH(generic-channel-name) +
    TYPE(USERMAP) CLNTUSER('client-user-name') +
    USRSRC(MAP) MCAUSER('user-id')
```

- **Implement rules such as:**
  - When 'wasadmin' connects, authorize as for ID 'myapp01'
  - When 'trob' connects, authorize as for ID 'mqm'
- **No wildcards in `CLNTUSER` attribute. One rule per ID.**
- **Many rules can specify the same target MCAUSER**
- **The same SVRCONN can map many IDs to 1 or more MCAUSER**

CHLAUTH BLOCKUSER is safer than mapping MCAUSER to a non-existent ID, as practiced in WMQ < v7.1.

Mapping QMgr Name

```
# Map one or more QMNAME to MCAUSER
SET CHLAUTH(generic-channel-name) +
    TYPE(QMGRMAP) QMNAME('generic-partner-qmgr-name') +
    USRSRC(MAP) MCAUSER('user-id')
```

- **Most effective when multiple QMgrs will use the same channel**
  - RCVR/RQSTR channels commonly have only one remote QMgr
    - Use mapping to detect change of partner QMgr
    - More typically, hard code MCAUSER to the appropriate value
  - Extremely useful for CLUSRCVR channels
    - We'll cover this case in detail later
- **Can include wildcards in QMNAME to map multiple partner nodes**
  - Useful when QMgr names follow structured pattern

Why Does Precedence Matter?

- **Enumerating all possible policies as individual rules would be impractical, if not entirely impossible**
- **Instead, establish generic rules that apply categorically, then enumerate the exceptions**
  - WMQ v7.1 standard policy is "deny all connections" with an exception for non-admin users on SYSTEM.ADMIN.SVRCONN
  - A base policy of deny-all with exceptions is preferred, and referred to as whitelisting
- **Two approaches:**
  - Rules are positional and processed in a strict order
  - Evaluate all rules in precedence order
- **WMQ takes the second of these approaches**

Not as Complicated as It Sounds

- **What is the value of x in the following calculation?**
  - 5 + 2 * 10 = x
- **By convention, the multiplication takes higher precedence than the addition so we process it as if it were written as follows:**
  - 5 + (2 * 10) = 25
- **Evaluation by precedence is more natural for humans than ordinal evaluation. It is also easier to work with when there are a great number of individual entries.**
- **The precedence of CHLAUTH rules was designed to be intuitive and similar to the precedence evaluation used by setmqaut and OAM.**

At What Points Does CHLAUTH Apply?

CHLAUTH:

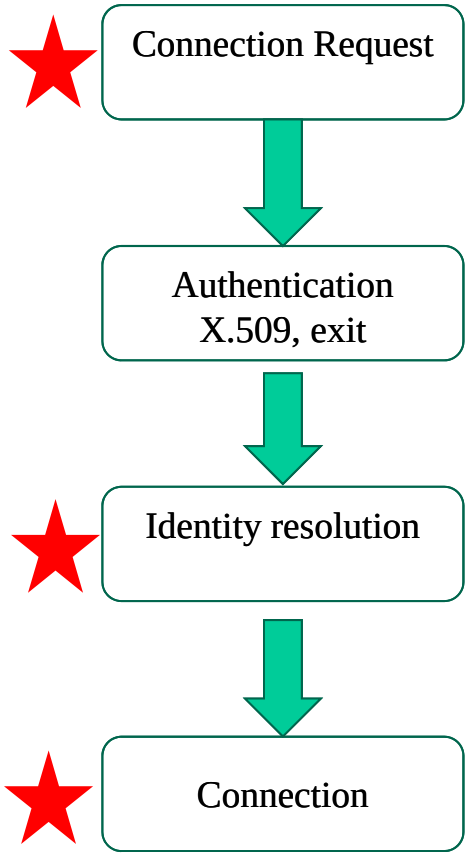
- ***Blocks*** **connections** ***at the listener*** **by IP address**
- ***Filters*** **connections by IP address** ***at the QMgr***

TLS/SSL makes an X.509 certificate available or channel.
Security exits may authenticate against LDAP, OS, etc.

CHLAUTH maps the following things to an MCAUSER:

- **Certificate distinguished name.**
- **The MCAUSER resolved by an exit.**
- **Client channel ID supplied by user or application.**
- **IP address.**

CHLAUTH: Final check of resolved MCASER .



Agenda

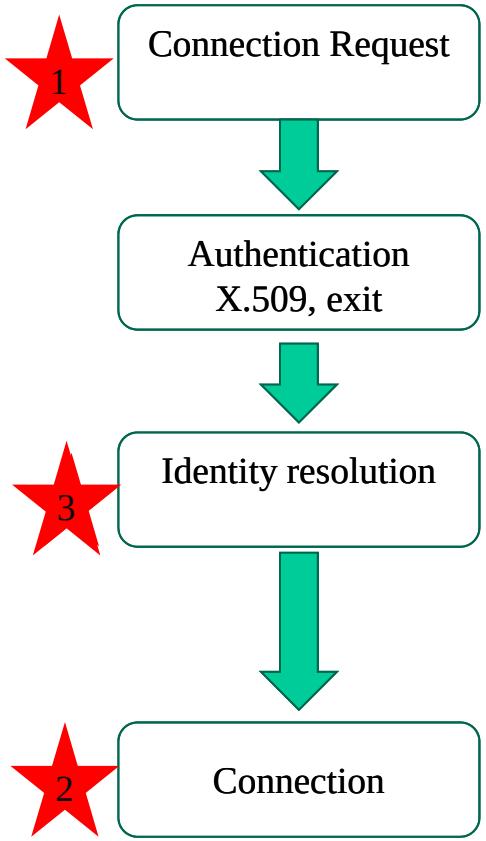
- WebSphere MQ Architecture
- Deep dive into CHLAUTH rules
- CHLAUTH precedence
- CHLAUTH use case: Granular cluster security
- Managing certificate expiry
- Other new GSKit 8 features
- Additional Resources

Functional CHLAUTH Precedence

1. **Blocking connections at the listener overrides all other rules because they are processed in the QMgr. If the connection doesn't make it to the QMgr there is no way to evaluate any other rules.**

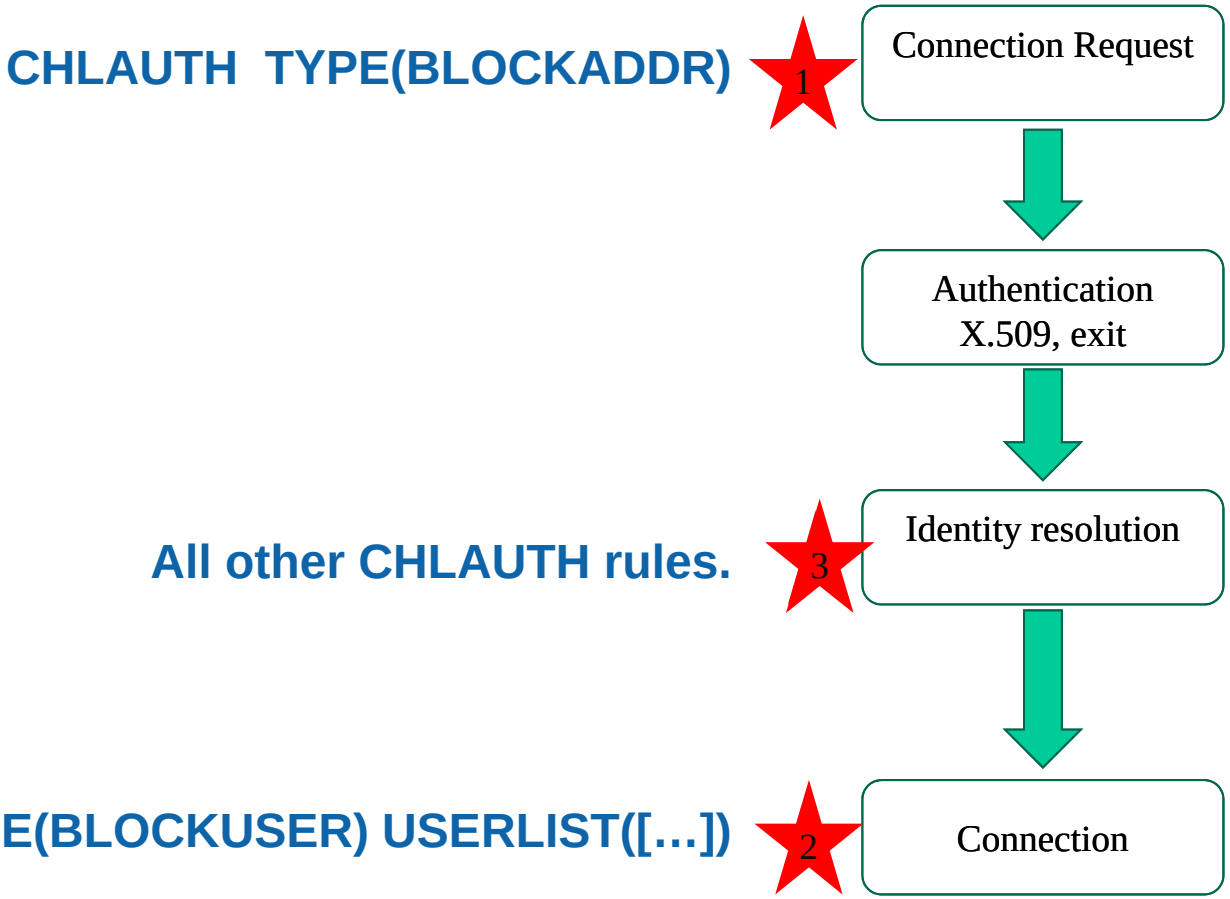
3. **The rules processed next decide whether to block the channel based on authentication criteria and they map identities. However, there is one last check which may yet block the channel.**

2. **Once all authentication and mapping rules and any channel exits have run, one last check is performed against the resolved ID in the MCAUSER.**



Note that the order of precedence differs from the order of execution! The final check can veto prior approval.

Functional CHLAUTH Precedence



CHLAUTH Precedence in a Nutshell

Extremely simple and intuitive:

The more specific,
the higher the precedence.

There are a number of rule types and attributes that come into play but after all is said and done, this one principle is all there is to CHLAUTH precedence. Simple. Intuitive.

Precedence of Different CHLAUTH Rule Types

When rules of different types match, they are valuated in the following order:

Order	Identity mechanism
0	Channel name
1	Certificate Distinguished Name
2=	Client asserted User ID
2=	Remote Queue Manager Name
4	IP address of remote node

Because all attributes can apply to multiple channels…
and a certificate is more unique than an ID or QMgr name…
and the ID and QMgr name are more unique than an IP address.

Simple. Intuitive.

Precedence for Multiple Rules of the Same Type

As expected, when there are two matches for a given rule type, the most specific one matches

- **In general, the further to the right the wildcard appears, the more specific is the rule**
- **Where the IP address can be in the middle of the attribute, the one with the most non-wild characters wins**

10.1.*.3 is more specific than 10.1.*
10.1.*.3 is more specific than 10.*.2.3
10.1.2.3 is more specific than any of the above
All of the above would match an IP of 10.1.2.3

String matching works the same. Simple. Intuitive.

Certificate DN Matching

Certificate Distinguished Names have many attributes but the precedence mapping is still based on specificity.

Order	DN substring	Name
1	SERIALNUMBER=	Certificate serial number
2	MAIL=	Email address
3	E=	Email address (Deprecated in preference to MAIL)
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 or province name
13	PC=	Postal code / zip code
14	C=	Country
15	UNSTRUCTUREDNAME=	Host name
16	UNSTRUCTUREDADDRESS=	IP address
17	DNQ=	Distinguished name qualifier

A Note About Organizational Units

1. **If they have different numbers of OU attributes, then the DN with the most OU values is more specific. This is because the DN with more Organizational Units fully qualifies the DN in more detail and provides more matching criteria.**
   - Even if its top-level OU is a wildcard (OU=*), the DN with more OUs is still regarded as more specific overall
2. **If they have the same number of OU attributes then the corresponding pairs of OU values are compared in sequence left-to-right, where the left-most OU is the highest-level (least specific), according to the following rules:**
   a. An OU with no wildcard values is the most specific because it can only match exactly one string
   b. An OU with a single wildcard at either the beginning or end (for example, OU=ABC* or OU=*ABC) is next most specific
   c. An OU with two wildcards for example, OU=*ABC*) is next most specific
   d. An OU consisting only of an asterisk (OU=*) is the least specific

CHLAUTH Precedence Recap

- **The most specific match takes precedence**
  - Across types, the most specific type wins
  - Across rules of the same type, most specific attribute wins
  - The OU attribute of a DN may occur more than once
    - Recall that the OU attributes are expected to describe a hierarchy, starting with the most generic element
      - Therefore rightmost elements are more specific
    - More OU elements wins over less, even with a wildcard

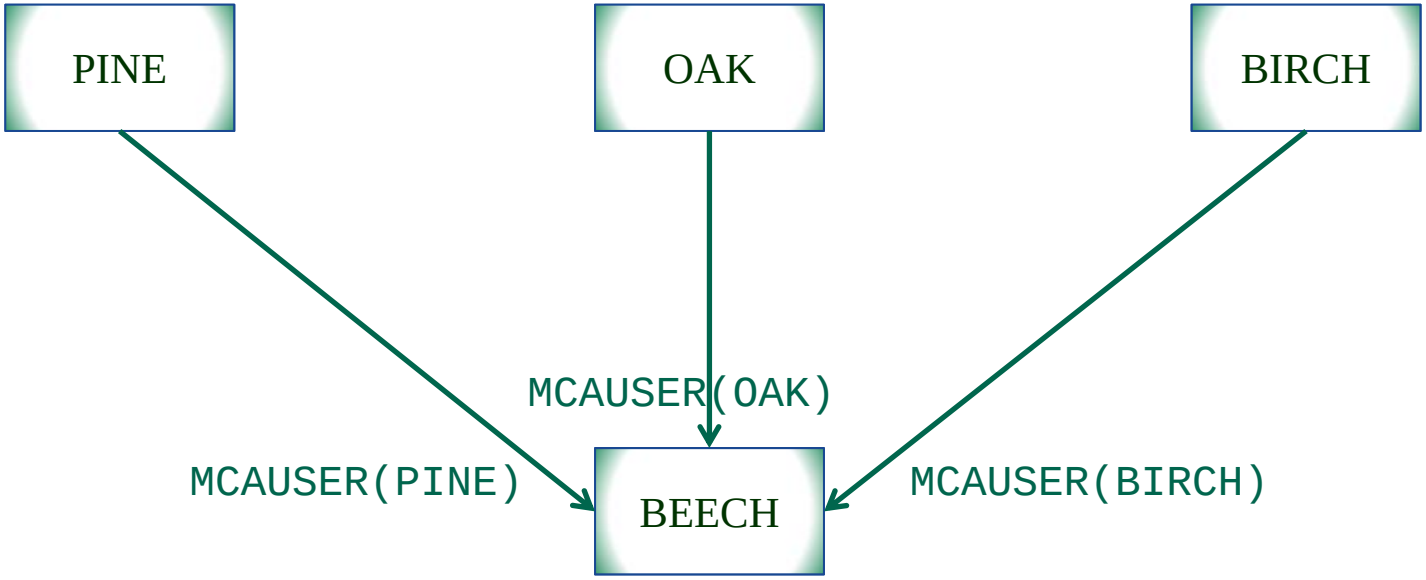
To understand CHLAUTH precedence
is to understand CHLAUTH.

Agenda

- WebSphere MQ Architecture
- Deep dive into CHLAUTH rules
- CHLAUTH precedence
- CHLAUTH use case: Granular cluster security
- Managing certificate expiry
- Other new GSKit 8 features
- Additional Resources

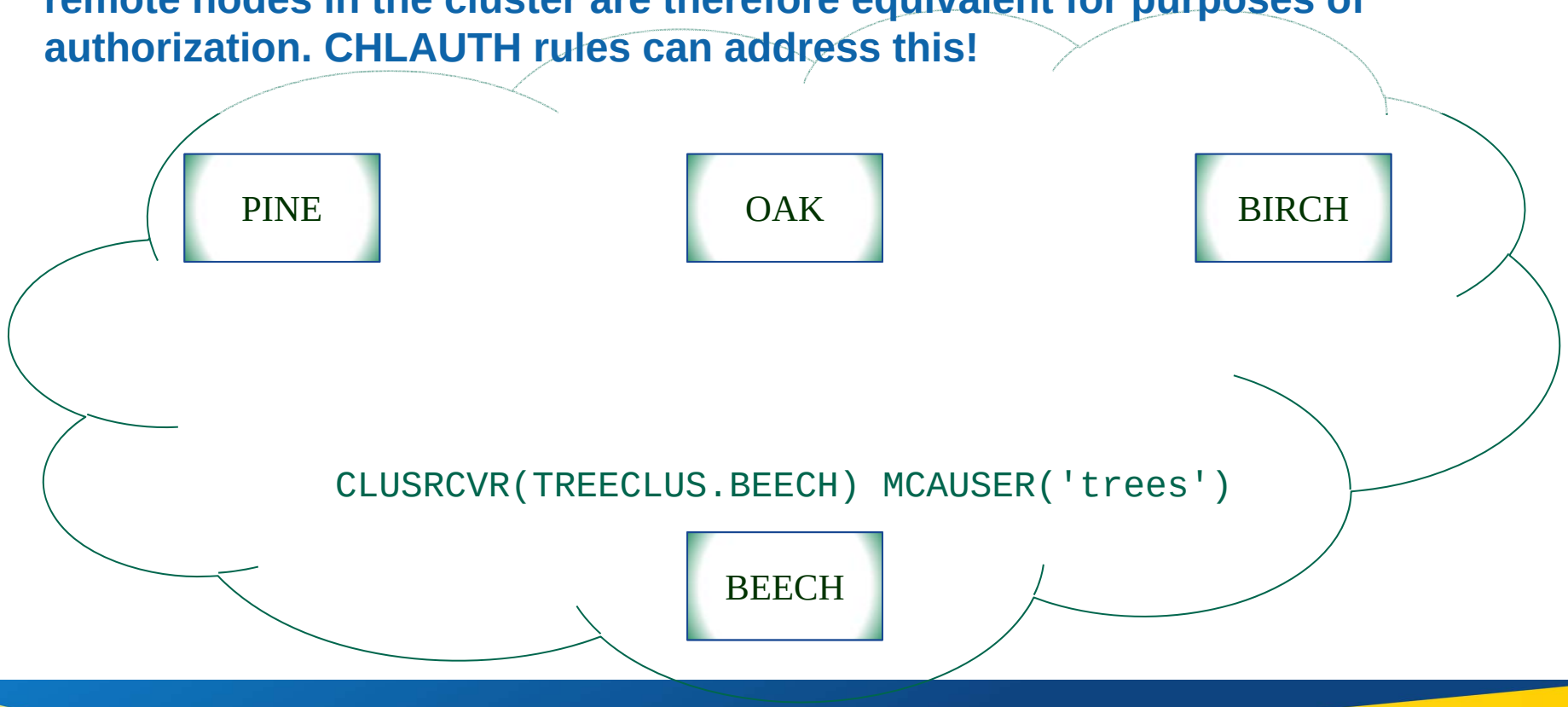
Authorization in P2P Networks

In non-clustered networks, each remote node has a dedicated channel. Each of these channels can contain a different MCAUSER value. It is therefore possible to grant different authorizations to each remote queue manager.



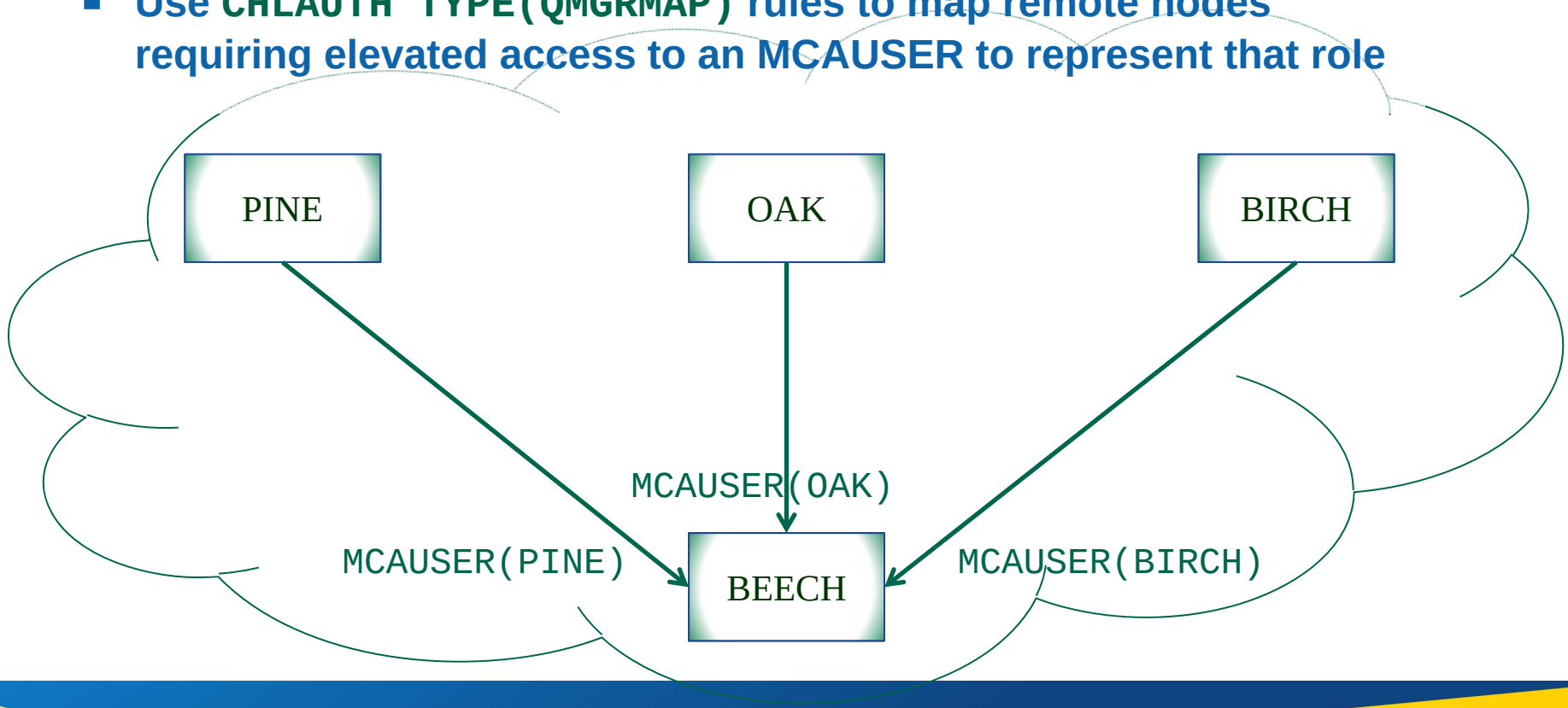
Authorization in WebSphere MQ Clusters < v7.1

In clustered networks all remote nodes use the same CLUSRCVR channel. Without exits the channel can contain only one value for MCAUSER. All remote nodes in the cluster are therefore equivalent for purposes of authorization. CHLAUTH rules can address this!



Authorization in v7.1+ WebSphere MQ Clusters

- Set a default MCAUSER value in the CLUSRCVR
- Use **`CHLAUTH TYPE(QMGRMAP)`** rules to map remote nodes requiring elevated access to an MCAUSER to represent that role



Example Cluster Authorization – Roles

- **The following roles have been identified for the CLUSRCVR channel:**
  - Repository – The full repositories for the cluster are hosted on dedicated queue managers
  - Cluster QMgrs – Each remote node represents a unique security role

- **The generic roles are not supported**
  - Administrators – Admins will connect directly, not through a remote queue manager
  - Guests – No unauthenticated or default access is defined. Only enumerated QMgrs in the cluster are recognized.

Example Cluster Authorization – Mapping

- **Each role is mapped to service account in a private group**
  - Repos Mapped to repos:repos
  - PINE Mapped to pine:pine
  - OAK Mapped to oak:oak
  - BIRCH Mapped to birch:birch

Example Cluster Authorization – Resources

- **The full repositories require limited access**
  - Cluster command queue
  - Dead Letter Queue
- **Cluster member queue managers require access to a specific subset of application queues**
  - Ordinary cluster QMgrs do not normally need access to:
    - Cluster transmit queue
    - Cluster command queue
    - SYSTEM.* queues in general

Example Cluster Authorization: Provision Access

Define the CLUSRCVR and map the remote queue managers:

```
# The MCAUSER(*NOACCESS) disables the channel.
DEF CLUSRCVR(TREECLUS.BEECH) CHLTYPE(CLUSRCVR) +
    CLUSTER(TREECLUS) MCAUSER(*NOACCESS) […]

# Mapping by QMgr enables the channel per remote node and
# sets MCAUSER of the running instance accordingly.
SET CHLAUTH(TREECLUS.BEECH) TYPE(QMGRMAP) +
    QMNAME(PINE)     MCAUSER('pine')
SET CHLAUTH(TREECLUS.BEECH) TYPE(QMGRMAP) +
    QMNAME(OAK)      MCAUSER('oak')
SET CHLAUTH(TREECLUS.BEECH) TYPE(QMGRMAP) +
    QMNAME(BIRCH)    MCAUSER('birch')
SET CHLAUTH(TREECLUS.BEECH) TYPE(QMGRMAP) +
    QMNAME('REPOS*') MCAUSER('repos')
```

Example Cluster Authorization: Repositories

Authorize the repositories

```
# Full repositories can connect to the QMgr…
setmqaut -m QMNAME -g repos -t qmgr        -all +connect +setall
# …and put to any SYSTEM queue…
setmqaut -m QMNAME -g repos -t queue \
         -n 'SYSTEM.**'                    –all +put +setall
# …except for the command queue.
setmqaut -m QMNAME -g repos -t queue \
         -n 'SYSTEM.ADMIN.COMMAND.QUEUE'   –all +none
# Here the DLQ is not a SYSTEM.* queue so needs another rule.
setmqaut -m QMNAME -g repos -t queue \
         -n 'BEECH.DEAD.LETTER.QUEUE'      –all +put +setall
```

Full repositories do not need broad access to SYSTEM.** queues but it simplifies the example.

Example Cluster Authorization: PINE

For purposes of this example, PINE needs to request services from the Customer application in the previous example

```
# PINE can connect to the QMgr…
setmqaut -m QMNAME -g pine -t qmgr        -all +connect +setall
# …and the DLQ…
setmqaut -m QMNAME -g pine -t queue \
         -n 'BEECH.DEAD.LETTER.QUEUE'     –all +put +setall
# …and request services.
setmqaut -m QMNAME -g pine -t queue \
         -n 'CUST.**.RQST'                 –all +put +setall
```

Example Cluster Authorization: OAK

For purposes of this example, OAK is allowed to request updates from the Customer application

```
# PINE can connect to the QMgr…
setmqaut -m QMNAME -g oak -t qmgr       -all +connect +setall
# …and the DLQ…
setmqaut -m QMNAME -g oak -t queue \
         -n 'BEECH.DEAD.LETTER.QUEUE'     –all +put +setall
# …and request services…
setmqaut -m QMNAME -g oak -t queue \
         -n 'CUST.**.RQST'                –all +put +setall
# …and request updates.
setmqaut -m QMNAME -g oak -t queue \
         -n 'CUST.**.UPDT'                –all +put +setall
```

Example Cluster Authorization – Review

- **Break down cluster security into roles**
- **Map one or more QMgrs to a role**
- **Authorize to the required subset of queues**
  - Full repositories are the only queue managers that need to write to the cluster command queue
  - As a rule, grant access to the DLQ for all remote nodes
  - As a rule, no remote node should require access to …
    - A transmit queue (all cluster traffic is 1-hop)
    - The system command queue

Agenda

- WebSphere MQ Architecture
- Deep dive into CHLAUTH rules
- CHLAUTH precedence
- CHLAUTH use case: Granular cluster security
- Managing certificate expiry
- Other new GSKit 8 features
- Additional Resources

New features in GSKit 8 support key expiry reporting

- New `-expire n` option for listing certificates in a keystore
- Takes numeric argument representing a number of days
- Any certificates which will expire in less than that number of days are highlighted in the listing by printing their validity dates
- The following examples were generated using certificates set to expire in 10, 30 and 60 days, all in the same keystore
- For convenience, the certificate labels in the examples that follow represent the certificate expiry threshold

Certificates expiring in 20 days or less

```
D:\tmp>runmqakm -cert -list -db key.kdb -stashed -expiry 20
5724-H72 (C) Copyright IBM Corp. 1994, 2011.  ALL RIGHTS
RESERVED.
Certificates found
* default, - personal, ! trusted
-       30-Day Expiry
*-      10-Day Expiry
    Not Before : September 21, 201311:27:26 AM EDT
    Not After : October 2, 201311:27:26 AM EDT
-       60-Day Expiry
```

Certificates expiring in 40 days or less

```
D:\tmp>runmqakm -cert -list -db key.kdb -stashed -expiry 40
5724-H72 (C) Copyright IBM Corp. 1994, 2011.  ALL RIGHTS
RESERVED.
Certificates found
* default, - personal, ! trusted
-       30-Day Expiry
    Not Before : September 21, 201311:27:46 AM EDT
    Not After : October 22, 201311:27:46 AM EDT
*-      10-Day Expiry
    Not Before : September 21, 201311:27:26 AM EDT
    Not After : October 2, 201311:27:26 AM EDT
-       60-Day Expiry
```

Certificates expiring in 60 days or less

```
D:\tmp>runmqakm -cert -list -db key.kdb -stashed -expiry 80
5724-H72 (C) Copyright IBM Corp. 1994, 2011.  ALL RIGHTS
RESERVED.
Certificates found
* default, - personal, ! trusted
-       30-Day Expiry
    Not Before : September 21, 201311:27:46 AM EDT
    Not After : October 22, 201311:27:46 AM EDT
*-      10-Day Expiry
    Not Before : September 21, 201311:27:26 AM EDT
    Not After : October 2, 201311:27:26 AM EDT
-       60-Day Expiry
    Not Before : September 21, 201311:28:13 AM EDT
    Not After : November 1, 201311:28:13 AM EDT
```

Agenda

- WebSphere MQ Architecture
- Deep dive into CHLAUTH rules
- CHLAUTH precedence
- CHLAUTH use case: Granular cluster security
- Managing certificate expiry
- Other new GSKit 8 features
- Additional Resources

TLS for authentication only

- **Prior to WMQ V7.1, the lightest weight channel cipherspec always hashed messages to provide integrity**
  - Imposed a slight CPU penalty
  - Even when TLS/SSL was only used for authentication
- **New with WMQ V7.1 and GSKit V8 is a NULL/NULL cipherspec**
  - Specified as TLS_RSA_WITH_NULL_NULL
  - TLS handshake, falls back to plaintext channel
- **Be sure to set SSLCAUTH(REQUIRED) on the channel!**

Empty keystores

- **In WMQ prior to V7.1, a new KDB keystore was populated with several commercial root and intermediate certificates**
- **This required the administrator to delete the unused certificates in order to prevent any possibility of distinguished name collisions**
- **As of WMQ V7.1, any new KDB keystores are created empty**
- **A new command has been added to populate the keystore from a set of known CA certificates**
  - During initial keystore creation
  - Or as a conversion option
- **Still possible to import CA signer certificates individually as before**

Support for new standards, removal of others

- **As of GSKit 8:**
  - SHA-2 family of algorithms supported for channel cipherspecs
    - Sha-2 for cert signing was supported in GSKit 7
  - Elliptic Curve ciphers supported for channel cipherspecs
  - Suite-B and updated FIPS standards supported
  - Some channel cipherspecs were disqualified for use with QMgr FIPS(ENABLED)
    - The word FIPS is in the name of these cipherspecs, however NIST has disqualified them under the new standard.
    - The disconnect between the standard and the cipherspec name can cause confusion

Agenda

- WebSphere MQ Architecture
- Deep dive into CHLAUTH rules
- CHLAUTH precedence
- CHLAUTH use case: Granular cluster security
- Managing certificate expiry
- Other new GSKit 8 features
- Additional Resources

Additional Resources

- **http://ibm.co/SETCHLAUTH**
  - Infocenter page for the SET CHLAUTH command
- **http://ibm.co/CHLAUTHPrecedence**
  - Infocenter page describing CHLAUTH records and precedence
- **http://ibm.co/DISCHLAUTH**
  - Infocenter page for DIS CHLAUTH with MATCH(RUNCHECK)
- **http://ibm.co/qminiSecurity**
  - Infocenter page describing the Security stanza of the qm.ini file

Questions & Answers