Channel Authentication Records
A New Channel Security feature in WebSphere MQ v7.1

Morag Hughson - hughson@uk.ibm.com

IBM UK

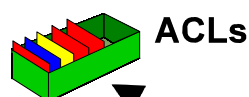**Session # 12612**

Channel Authentication Records

- **Set rules to control how inbound connections are treated**
  - ▶ Inbound Clients
  - ▶ Inbound QMgr to QMgr channels
  - ▶ Other rogue connections causing FDCs

- **Rules can be set to**
  - ▶ Allow a connection
  - ▶ Allow a connection and assign an MCAUSER
  - ▶ Block a connection
  - ▶ Ban privileged access
  - ▶ Provide multiple positive or negative SSL Peer Name matching

- **Rules can use any of the following identifying characteristics of the inbound connection**
  - ▶ IP Address
  - ▶ SSL/TLS Subject's Distinguished Name
  - ▶ Client asserted user ID
  - ▶ Remote queue manager name

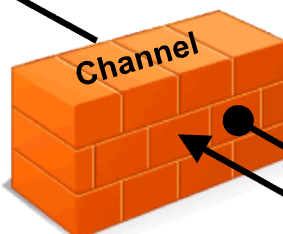# Channel Authentication Records – Notes

N

O

T

E

S

- Channel Authentication records allow you to define rules about how inbound connections into the queue manager should be treated. Inbound connections might be client channels or queue manager to queue manager channels. These rules can specify whether connections are allowed or blocked. If the connection in question is allowed, the rules can provide a user ID that the channel should run with or indicate that the user ID provided by the channel (flowed from the client or defined on the channel definition) is to be used.
- These rules can therefore be used to
  - Set up appropriate identities for channels to use when they run against the queue manager
  - Block unwanted connections
  - Ban privileged users
- Which users are considered privileged users is slightly different depending on which platform you are running your queue manager on. There is a special value '*MQADMIN' which has been defined to mean "any user that would be privileged on this platform". This special value can be used in the rules that check against the final user ID to be used by the channel – TYPE(USERLIST) rules – to ban any connection that is about to run as a privileged user. This catches any blank user IDs flowed from clients for example.
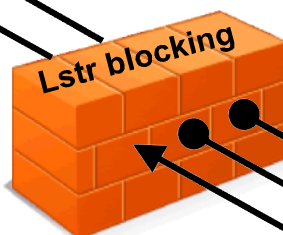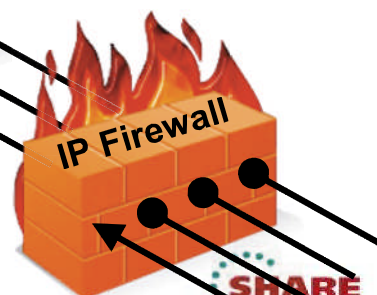
Channel Access Blocking Points

ACLs

■ **Channel Blocking/Mapping**
- ▶ Rules to block channels
- ▶ Rules to map channels to MCAUSER
- ▶ Rules to allow channels as they are
- ▶ Runs before security exit
- ▶ Final check for user ID before allowing through
  - • After Security Exit has run and final MCAUSER is assigned
  - • Ban privileged users with '*MQADMIN'

Channel

Lstr blocking

IP Firewall

■ **Listener Blocking**
- ▶ NOT A REPLACEMENT FOR AN IP FIREWALL!!
- ▶ Blocked before any data read from the socket
- ▶ Simplistic avoidance of DoS attack
  - • Really the place of the IP firewall
- ▶ Network Pingers if blocked don't raise an alert

SHARE in San Francisco 2013

Channel Access Blocking Points – Notes

N

- ▪ In this picture we illustrate that there are a number of points that an inbound connection must get through in order to actually make use of an MQ queue.
- ▪ First, we remind you that your IP firewall is included in this set of blocking points and should not be forgotten, and is not superseded by this feature in MQ.

O

- ▪ One point of note, the inbound connections can be from any version of MQ. There is no requirement that the clients or remote queue managers also be on WebSphere MQ V7.1 to be blocked or mapped by these rules.

T

E

S

SHARE in San Francisco 2013

Block Point: MQ Listener

- **List of IP address patterns**

- **One single list**

> **SET CHLAUTH(*) TYPE(BLOCKADDR)**
> **ADDRLIST('9.20.*', '192.168.2.10')**

- **NOT A REPLACEMENT FOR AN IP FIREWALL**
  - ▶ Temporary blocking
  - ▶ Blocking until IP firewall updated
  - ▶ Shouldn't be many entries in the list

- **Blocked before any data read from the socket**
  - ▶ i.e. before SSL Handshake
  - ▶ Before channel name is known

- **Avoiding DoS attack**
  - ▶ Really the place of the IP firewall
  - ▶ Simplistic 'hold' of inbound connection to avoid reconnect busy loop

- **Network Pingers if blocked don't raise an alert**
  - ▶ Immediate close of socket with no data not considered a threat

SHARE
in San Francisco
2013

Channel Access Blocking Points – Notes

N O T E S

- Second, there is a list of IP addresses that the listener configuration will have access to. If any of these IP addresses attempts to start an inbound channel connection, the listener will bounce the connection prior to starting a channel instance to process any data, for example SSL Handshake, that might need to be read before knowing what the channel name is. If the queue manager is not running it will still have access to the configuration and it will still block the specified IP addresses. THIS IS NOT A REPLACEMENT FOR AN IP FIREWALL! However, it does provide a way for an MQ Administrator to implement temporary blocking until the IP firewall updated, or for a short period of time making it not worthwhile to update the IP firewall. The intention is that there shouldn't be many entries in the list.

- A Denial of Service (DoS) attack on the listener; whilst really the place of the firewall to deal with; would mean high CPU in the listener if it had to deal with a repeated connection from a inbound connection. This and the fact that we would like to quietly ignore network pingers if they don't send any data and only raise blocking events on them if they do send data, means that the listener will hold any sockets that come from blocked IP address open for a period of time prior to closing the socket, rather than immediately rejecting it. This will stall the repetitiveness of the attack and protect the listener process allowing it some time to process real requests, and additionally give the network pingers time to close the socket before we do allowing us to detect we don't need to emit an event for that case. By default this time will be 30 seconds.

- Thirdly we come to the rules that work on specific channels. You can set up rules to match against all of the identifying characteristics of an inbound channel (see next notes page). These rules can either indicate that a channel matching the rule should be blocked; should be allowed and assigned a provided user ID to use when it runs; or allowed and the user ID provided by the channel is to be used.

SHARE
in San Francisco
2013

Mapping Point: MQ Channel

- **Duplets of identifying attribute mapped to MCAUSER**

- **Identifying attributes are Channel Name and**
    - ▶ SSL Peer Name pattern (most specific)
        - Precedence defined for partial patterns
    - ▶ Remote queue manager name pattern (MCA channels)
    - ▶ Client asserted user ID (MQI channels)
        - No pattern matching on this
    - ▶ IP address pattern (least specific)

- **Mapping done before calling security exit**

- **Parameter to indicate where user ID is taken from**
    - ▶ Provided on command
    - ▶ Flowed or defined on channel as today
        - Combine this with BLOCKUSER list
    - ▶ Blocked

Order	DN Substring	Name
1	CN=	Common name
2	T=	Title
3	OU=	Organizational unit
4	O=	Organization
5	L=	Locality
6	ST=, SP=, S=	State or province name
7	C=	Country

Order	Identity mechanism	Notes
0	Channel Name	
1	SSL Distinguished Name	
2=	Client asserted User ID	Clearly several different user IDs can be running on the same IP address.
2=	Queue Manager Name	Clearly several different queue managers can be running on the same IP address
4	IP address	

Channel Access Blocking Points – Notes

N O T E S

IP Address
- Rules can be made to be used should a connection arrive from the specified IP address.
- If the client asserted a banned user ID, but its IP address is in this list to map it to another user ID it will be deemed to have asserted the mapped user ID and not the banned one, so the problem of an older Java client sending up blank can be fixed by mapping and it doesn't have to end up banned. In other words mapping happens before the blocked list of user IDs is checked. If you have a port forwarder, DMZ session break, or any other setup which will change the IP address presented to the queue manager, then mapping IP addresses is not necessarily suitable for your use.
- The patterns that can be used to specify IP addresses are described later. Additionally, we will find the most specific rule in order to do the mapping, so an additional pattern is allowed in this configuration – that is a single asterisk which means 'match everything'.

SSL/TLS DN
- Rules can be made to be used should a connection arrive from the specified DN.
- If the client asserted a banned user ID, but its DN is in this list to map it to another user ID it will be deemed to have asserted the mapped user ID and not the banned one in the same way as described above.
- DNs can be provided with pattern matching in the same way as the SSLPEER attribute that already exists. That defines the match everything pattern to be "CN=*". As for IP addresses, we will find the most specific duplet in order to do the mapping; this requires us to define a precedence order of substrings in the DN in order to decide which to use if we have two DN patterns that both match the full DN. An example will follow.

Remote Queue Manager Name
- Rules can be made to be used should a connection arrive from the specified remote queue manager.
- There is also pattern matching on queue manager names, although it is by its very nature, much simpler patterns than IP addresses or SSL Peer Names.

Client asserted User ID
- Rules can be made should a connection arrive asserting the specified user ID. It is often likely that the client asserted user ID is not even defined on the server system, so this can be used to map it to an MCAUSER user ID which of course will be a server side defined user ID.
- If the client asserted a banned user ID, but that user ID is in this to map it to another user ID it will be deemed to have asserted the mapped user ID and not the banned one in the same way as the other mappings.
- There is a list of user IDs which, if any of them are asserted, will cause the channel to end. The checking against this list of blocked user IDs runs after all things that can set the MCAUSER have completed, including security exits and the MCAUSER mapping feature we just spoke of. *MQADMIN is a recognition that we logically want to ban the 'mqm' user if you like. However, this is actually a differing set of user IDs dependant on platform, so we will discover at run-time whether we have a privileged user, and if *MQADMIN is blocked, then privileged users are blocked.
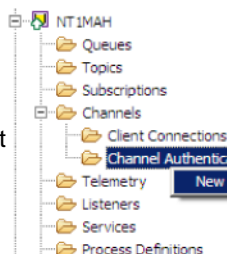
Channel Authentication Records – Configuration

- **Create rules using**
  - MQSC: SET CHLAUTH
  - PCF
  - MQ Explorer GUI Wizard

- **Pattern matching**
  - Channel Name
    - Beginning, middle, end
  - IP addresses (IPV4 or IPV6)
    - '*' in any segment
    - '-' ranges in any segment
  - SSL Peer Name (as today)
  - QMgr Name – as channel name

- **Restricting rules further by IP Address**
  - Rules matching on
    - SSL Peer Name
    - Remote QMgr Name
    - Client User ID
  - Can add IP address

- **Precedence matching**
  - Most specific rule is matched
  - Within SSL Peer Name matching
    - Most specific substring is matched

```
Command Prompt - runmqsc TEST1

Starting MQSC for queue manager TEST1.

SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)

SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('9.20.1-3.*')
USERSRC(CHANNEL)

SET CHLAUTH('SYSTEM.ADMIN.*') TYPE(SSLPEERMAP) SSLPEER('O=IBM')
USERSRC(CHANNEL)

SET CHLAUTH('QM1.TO.QM2') TYPE(QMGRMAP) QMNAME(QM1)
USERSRC(MAP) MCAUSER('QM1USER')

SET CHLAUTH('*.SVRCONN') TYPE(USERMAP)
CLNTUSER('mhughson') MCAUSER('hughson@hursley')

SET CHLAUTH('*') TYPE(SSLPEERMAP) SSLPEER('CN="Morag Hughson"')
ADDRESS('9.*') MCAUSER('hughson')
```

Tree (MQ Explorer):
- NT1MAH
  - Queues
  - Topics
  - Subscriptions
  - Channels
    - Client Connections
    - Channel Authentication Records
      - New ▶ Channel Auth
  - Telemetry
  - Listeners
  - Services
  - Process Definitions

Chl: MY.CHANNEL
IP: 9.20.1.123
DN: CN=Morag Hughson.O=IBM UK
UID: mhughson

Order	Identity mechanism
0	Channel Name
1	SSL Distinguished Name
2=	Client asserted User ID
2=	Queue Manager Name
4	IP address

2013

Channel Authentication – Configuration – Notes

N O T E S

- Here we show some example rules illustrating the commands used for creating the rules. These examples are in MQSC. There is also PCF, and this is used by the MQ Explorer GUI. Additionally, the MQ Explorer GUI provides a wizard to walk you through the steps for setting up these rules and at the end of the wizard, the MQSC command that would do the same job as you have done in the wizard, is displayed in a window that you can cut'n'paste from to put the command into a script for future use.
- Some of these examples illustrate the pattern matching that can be applied to channel names, IP addresses, SSL/TLS DNs and remote queue manager names. Also we see all three types of rules, blocking channels – USERSRC(NOACCESS); allowing channels to run with the user ID provided by the channel – USERSRC(CHANNEL); and assigning a user ID to a channel – USERSRC(MAP) MCAUSER(user-id). USERSRC(MAP) is the default so we also see in another example that it does not need to be specified on the command.
- When mapping from an SSL certificate DN you may also want to ensure that certificate is being used from the correct IP address, mitigating what might happen if a certificate is stolen.
- When mapping from a queue manager name, you may also want to ensure that the queue manager is running on the correct IP address to ensure it is not a rogue queue manager with the same name as one in your cluster for example.
- When there is more than one rule that could match the inbound connection in question, then we define which rule will actually be used by defining the precedence order of what is the most specific match. The table shows that SSL Peer Names are considered a more specific match than a queue manager name or client user ID (because there is much more detailed information in an SSL Peer Name); and IP addresses are considered the least specific since clearly more than one queue manager or client can be connecting from the same IP address.

SHARE in San Francisco
2013

Restricting the Mappings

- **Restrict where an SSL Certificate can be used from**
  - ▶ Specific IP address

- **Restrict where a queue manager or client user ID can come from**
  - ▶ Specific IP address

Mapped \ Restrict By	SSL Peer	QM Name	Client User	IP Address
SSL Peer		X	X	✓
QM Name				✓
Client User				✓
IP Address				

```
SET CHLAUTH(*) TYPE(SSLPEERMAP)
SSLPEER('L="Hursley"') MCAUSER(HURUSER) ADDRESS('9.20.*')


SET CHLAUTH(*) TYPE(QMGRMAP)
QMNAME(CLUSQM*) MCAUSER(CLUSUSR) ADDRESS('9.30.*')
```

Restricting the Mappings - Notes

N
O
T
E
S

- When mapping from an SSL certificate DN, you may also want to ensure that certificate is being used from the correct IP address, mitigating what might happen if a certificate is stolen.
- When mapping from a queue manager name, you may also want to ensure that the queue manager is running on the correct IP address to ensure it is not a rogue queue manager with the same name as one in your cluster for example.
- We could imagine using the remote queue manager name or the client user ID as a restrictor on an SSL Peer rule, however feedback from EAP did not suggest anyone needed it so it was not implemented. For the most part, attributes within the X509 DN will contain the same information for most practical uses. For example CN=<Queue Manager Name>.

SSL DN Precedence Mapping Example

SET CHLAUTH(*) TYPE(SSLPEERMAP) SSLPEER('OU="MQ Devt"')
MCAUSER(MQUSER)

SET CHLAUTH(*) TYPE(SSLPEERMAP) SSLPEER('L="Hursley"')
MCAUSER(HURUSER)

Order	DN Substring	Name
1	CN=	Common name
2	T=	Title
3	OU=	Organizational unit
4	O=	Organization
5	L=	Locality
6	ST=, SP=, S=	State or province name
7	C=	Country

Most Specific Match

Certificate
CN=Mor...
Hughs

CN=Morag Hughson.OU=MQ Devt.
O=IBM UK.L=Hursley.C=UK

SSL DN Precedence Mapping Example – Notes

NOTES

- Not only do we define the order of precedence between the various different identifying characteristics of an inbound connection, we also must do a similar job for SSL Peer Name.
- Here is an example to illustrate what happens when two partial patterns could both match an inbound Distinguished Name (DN) from a client.
- We want the most specific match to be used, so we have defined a precedent order of what we mean by the most specific.
- The table shown here that defines the precedence order is a subset of the contents of an SSL Peer Name in WebSphere MQ V7.1. It suffices to describe this example. For the full table of SSL Peer Name attributes, search the MQ Information Centre for "Distinguished Names".

IP Address Pattern Matching

- **Single Address**
- **Wildcard at the end**
- **Wildcard in the middle**
- **Ranges**
- **IPV4 or IPV6**
- **IPV6 wildcarded**
- **IPV4 will also block IPV6 and vice versa**

- **9.20.4.6**
- **9.20.***
- **9.20.*.6**
- **9.20.4.1-10**
- **3ffe:1900:4545:3:200:f8ff:fe21:67cf**
- **3ffe:1900:4545:3:200:***
- **0:0:0:0:0:ffff:192.1.56.10**

IP Address Pattern Matching – Notes

N
O
T
E
S

- The IP addresses can be specified as single addresses, e.g. 9.20.4.6 or as patterns, e.g. 9.20.* which would of course also match the former. There patterns can also be generic in the middle, not just at the end, e.g. 9.20.*.6; and can provide ranges (rather akin to how you might configure a firewall) e.g. 9.20.4.1-20.
- These patterns of course will also understand IPV6 address, so as another example one might provide 3ffe:1900:4545:3:200:f8ff:fe21:67cf or 3ffe:1900:4545:3:200:* which would also match the specific address. We must also understand that 0:0:0:0:0:ffff:192.1.56.10 is the same as 192.1.56.10 so that the correct refusals are made when IPV6 and IPV4 are both in use.
- Hostnames cannot be specified in this list – only IP addresses. Firewalls only seem to operate in IP addresses too and this is so similar in nature to a firewall. Also, hostnames can be different/unknown depending on what end of the channel you are at, after all it is only essential for the initiating end to even be able to resolve the hostname.

How should I use this?

```
SET CHLAUTH(*) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Shetland') MCAUSER(BANK123)

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Orkney') MCAUSER(BANK456)

SET CHLAUTH(SYSTEM.ADMIN.SVRCONN) TYPE(ADDRESSMAP)
ADDRESS('9.20.1-30.*') MCAUSER(ADMUSER)

SET CHLAUTH(TO.CLUS.*) TYPE(QMGRMAP)
QMNAME(CLUSQM*) MCAUSER(CLUSUSR) ADDRESS('9.30.*')
```

"Our internal cluster doesn't use SSL, but we must ensure only the correct queue managers can connect into the cluster"

How should I use this? - Notes

N
O
T
E
S

- Here is an example of how we expect this to be used.
- Our business requires that "We must make sure our system is completely locked down". So we start off with a rule that blocks everyone. Therefore anyone that doesn't match a more specific rule will not be allowed in.
- Our business requires that "Our Business Partners must all connect using SSL, so we will map their access from the certificate DNs". So we have some rules that map specific DNs of our Business Partners to specific user IDs. Previously you might have done this by having separate channel definitions for each BP, now if you wish they can come into the same receiver definition.
- Our business requires that "Our Administrators connect in using MQ Explorer, but don't use SSL. We will map their access by IP Address". So we have a rule that gives them all a single administrative access user ID based on a range of IP addresses.
- Our business requires that "Our internal cluster doesn't use SSL, but we must ensure only the correct queue managers can connect into the cluster". So we have a rule that gives access to the correctly named queue managers but only if they come from a recognised IP address.

Precedence Mapping Example

```
DISPLAY CHLAUTH(SYSTEM.*)
returns ===>
  CHLAUTH(SYSTEM.ADMIN.SVRCONN)
  TYPE(ADDRESSMAP)
  ADDRESS('9.180.165.163') MCAUSER(MORAG)

  CHLAUTH(SYSTEM.*)
  TYPE(USERMAP)
  CLNTUSER('mhughson') MCAUSER(HUGHSON)

  CHLAUTH(*)
  TYPE(SSLPEERMAP)
  SSLPEER('O="IBM UK"') MCAUSER(UKUSER)

  CHLAUTH(*)
  TYPE(ADDRESSMAP)
  ADDRESS('9.*') MCAUSER(IBMUSER)
```

MCAUSER(UKUSER)

Order	Identity mechanism
0	Channel Name
1	SSL Distinguished Name
2	Client asserted User ID
3	IP address

Chl: APPL1.SVRCONN
DN: CN=Morag Hughson.O=IBM UK
UID: mhughson
IP: 9.180.165.163

Precedence Mapping Example - Notes

N

O

T

E

S

- Here is an example to show what happens if a specific inbound channel could match a number of rules.
- First we recall that we earlier defined a most specific to least specific precedence matching rule for these identity attributes too. We show a basic version of that table here with just the relevant attributes for an inbound client channel.
- So the question is, if an inbound client channel came into this queue manager with the identifying attributes shown, which MCAUSER would it end up using?

- Imagine an inbound connection coming in over channel SYSTEM.ADMIN.SVRCONN with a certificate with SSL Distinguished Name of "CN=Morag Hughson, O=IBM UK", a client side user ID of "mhughson" and a source IP address of "9.180.165.163". This would match the first rule shown and provide MCAUSER of MORAG.
- Now imagine the same details, but coming in over channel SYSTEM.DEF.SVRCONN. This would match the second rule shown and provide MCAUSER of HUGHSON.
- Finally imagine the same details, but coming in over channel APP1.SVRCONN. This would match the third rule shown and provide MCAUSER of UKUSER.

- When displaying many rules the output of the display command will list them in precedence matching order to help in remembering these precedences.

- Since we recognise that this is a possible usability problem, we have also got a special flavour of the DISPLAY command where you can input the exact details of a likely inbound connection and be told what MCAUSER will be used should such a channel connect.

What happens if…?

```
DISPLAY CHLAUTH(SYSTEM.ADMIN.SVRCONN) MATCH(RUNCHECK)
         SSLPEER('CN="Morag Hughson", O="IBM UK"')
         CLNTUSER('mhughson')
         ADDRESS('9.180.165.163')
returns ===>
                CHLAUTH(SYSTEM.ADMIN.SVRCONN)
                TYPE(ADDRESSMAP)
                ADDRESS('9.180.165.163') MCAUSER(MORAG)
```

Chl: SYSTEM.ADMIN.SVRCONN
DN:  CN=Morag Hughson.O=IBM UK
UID: mhughson
IP:   9.180.165.163

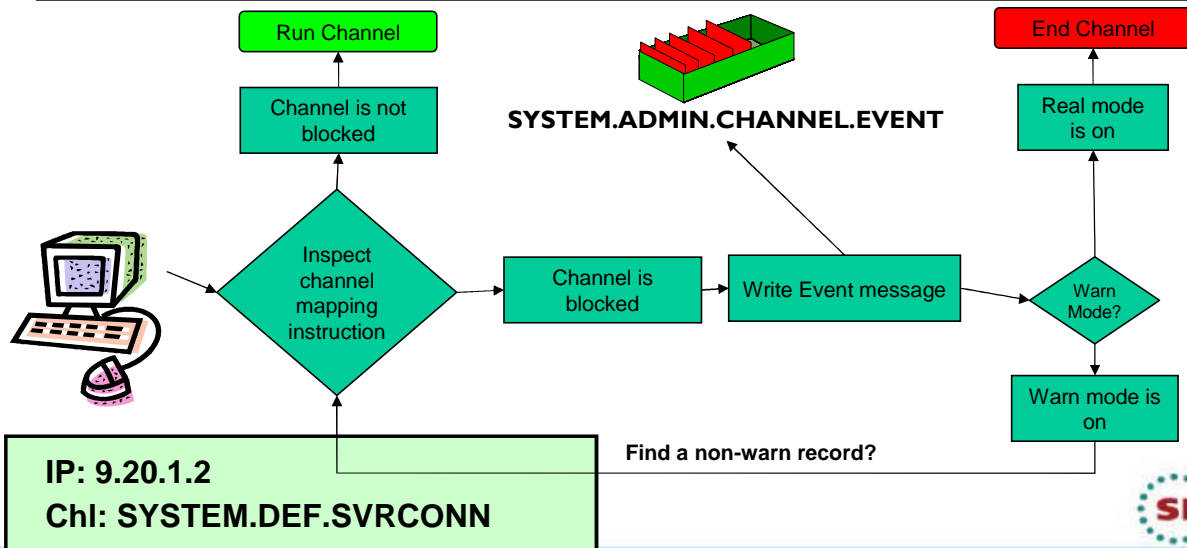| Order | Identity mechanism |
|-------|--------------------|
| 0 | Channel Name |
| 1 | SSL Distinguished Name |
| 2= | Client asserted User ID |
| 2= | Queue Manager Name |
| 4 | IP address |

---
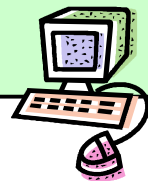
# What happens if…? - Notes

N
O
T
E
S

- Here is an example of the special matching version of the DISPLAY command to show exactly what would happen should a channel matching these identifying attributes, connect into the system. This should serve as a useful testing tool, service aid, and validation tool, although we would of course recommend not creating such complicated rules that you need it in the first place!

# Warning mode

```
ALTER QMGR CHLEV(ENABLED)

SET CHLAUTH(SYSTEM.*) TYPE(ADDRESSMAP) ADDRESS('9.20*') USERSRC(NOACCESS) WARN(YES)
SET CHLAUTH(SYSTEM.*) TYPE(ADDRESSMAP) ADDRESS('9.*') MCAUSER(HUGHSON)
SET CHLAUTH(SYSTEM.*) TYPE(ADDRESSMAP) ADDRESS(*) USERSRC(NOACCESS) WARN(NO)
```

Run Channel

Channel is not blocked

SYSTEM.ADMIN.CHANNEL.EVENT

End Channel

Real mode is on

Inspect channel mapping instruction → Channel is blocked → Write Event message → Warn Mode?

Warn mode is on

Find a non-warn record?

IP: 9.20.1.2
Chl: SYSTEM.DEF.SVRCONN

Warning Mode - Notes

N
O
T
E
S

- If the first record that matches an inbound channel's details is a blocking type of record and has the WARN field set to YES, the channel will cut an error and an event message to show that it would have matched that record, and then will look for the next matching record that does not have WARN set to YES to discover exactly what credentials it is going to use to run with. If the second record it finds is also a blocking type of record, then another error and event message will be written and the channel will end.

Out of the Box

- ■ **We supply these rules out-of-the-box.**
  - ▶ For all channels, ban the assertion of privileged users by inbound channels.
  - ▶ For all SYSTEM channels except SYSTEM.ADMIN.SVRCONN (the MQ Explorer GUI channel), ban anyone from using them.

```
SET CHLAUTH(*) TYPE(BLOCKUSER) USERLIST(*MQADMIN)

SET CHLAUTH(SYSTEM.*) TYPE(ADDRESSMAP)
ADDRESS(*) USERSRC(NOACCESS)

SET CHLAUTH(SYSTEM.ADMIN.SVRCONN) TYPE(ADDRESSMAP) ADDRESS(*)
USERSRC(CHANNEL)
```

  - ▶ Difficult to supply any default rules regarding IP addresses and SSL Peer Names since they are very installation specific.

- ■ **Enabling Switch ALTER QMGR CHLAUTH(ENABLED|DISABLED) different for Migrated or New Queue Manager**

---

Out of the Box - Notes

N

O

T

E

S

- ■ Out of the box we supply some rules.
- ■ The first is a rule which bans privileged users and blank users from being asserted by connecting inbound channels. This rule may break some channels, but it will secure many more channels than it breaks so we believe it to be a worthwhile out-of-the-box position.
- ■ The second rules secures the use of SYSTEM channels by disallowing any address from connecting. This stops hackers from connecting in to the SYSTEM.DEF.RECEIVER for example. It also locks down the SYSTEM.DEF.SVRCONN which will hit lots of people initially!
- ■ The third rule allows the SYSTEM.ADMIN.SVRCONN but it will still be affected by the first rule if you try to use a privileged user ID, so some work must be done to provide a user ID that has access to do what is needed.
- ■ There is a queue manager switch which determines whether CHLAUTH rules are acted upon (it does not stop the commands from be used though). This switch is ENABLED for new queue managers, and DISABLED for migrated queue managers.

Events

- **Command events (as normal)**

- **Configuration events (as normal)**

- **Channel event**

- **Controlled by existing switch**
  - Considered to be an EXCEPTION
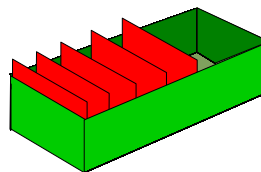
SYSTEM.ADMIN.CHANNEL.EVENT

- **Written to existing queue**

- **One event for each type of connection refusal**

> **ALTER QMGR CHLEV(ENABLED|EXCEPTION)**

- **MQRC_CHANNEL_BLOCKED**

  - MQRQ_CHANNEL_BLOCKED_ADDRESS
  - MQRQ_CHANNEL_BLOCKED_USERID
  - MQRQ_CHANNEL_BLOCKED_NOACCESS

Events - Notes

N O T E S

- These commands will generate command events and configuration events (assuming that these events are enabled by the existing CMDEV and CONFIGEV switches).

- There are some new events to record whenever an inbound connection attempt is blocked. Controlled by the current CHLEV switch (and considered to be an EXCEPTION) this new event message will be issued to the SYSTEM.ADMIN.CHANNEL.EVENT queue when a channel or listener blocks an attempt to connect.

- The reason qualifier of the event message can be

  - **MQRQ_CHANNEL_BLOCKED_ADDRESS**
    Channel was blocked due to its IP address being in the list to be refused.

  - **MQRQ_CHANNEL_BLOCKED_USERID**
    Channel was blocked due to its asserted (or mapped) user ID being in the list to be refused.

  - **MQRQ_CHANNEL_BLOCKED_NOACCESS**
    Channel was blocked due to its identity (e.g. IP address or SSL Peer name) being mapped to a rule that says it is to be blocked.

Channel Authentication Records - Recap

- **Set rules to control how inbound connections are treated**
  - ▶ Inbound Clients
  - ▶ Inbound QMgr to QMgr channels
  - ▶ Other rogue connections causing FDCs

- **Rules can be set to**
  - ▶ Allow a connection
  - ▶ Allow a connection and assign an MCAUSER
  - ▶ Block a connection
  - ▶ Ban privileged access
  - ▶ Provide multiple positive or negative SSL Peer Name matching

- **Rules can use any of the following identifying characteristics of the inbound connection**
  - ▶ IP Address
  - ▶ SSL/TLS Subject's Distinguished Name
  - ▶ Client asserted user ID
  - ▶ Remote queue manager name

Recap

N
O
T
E
S

- We saw this page at the beginning, but we will use it again as a summary. We have learned today how to use this new feature in WebSphere MQ V7.1 to control how our inbound connections will behave.