Secure Universal Messaging: Five MQ 7.1 Security Use Cases

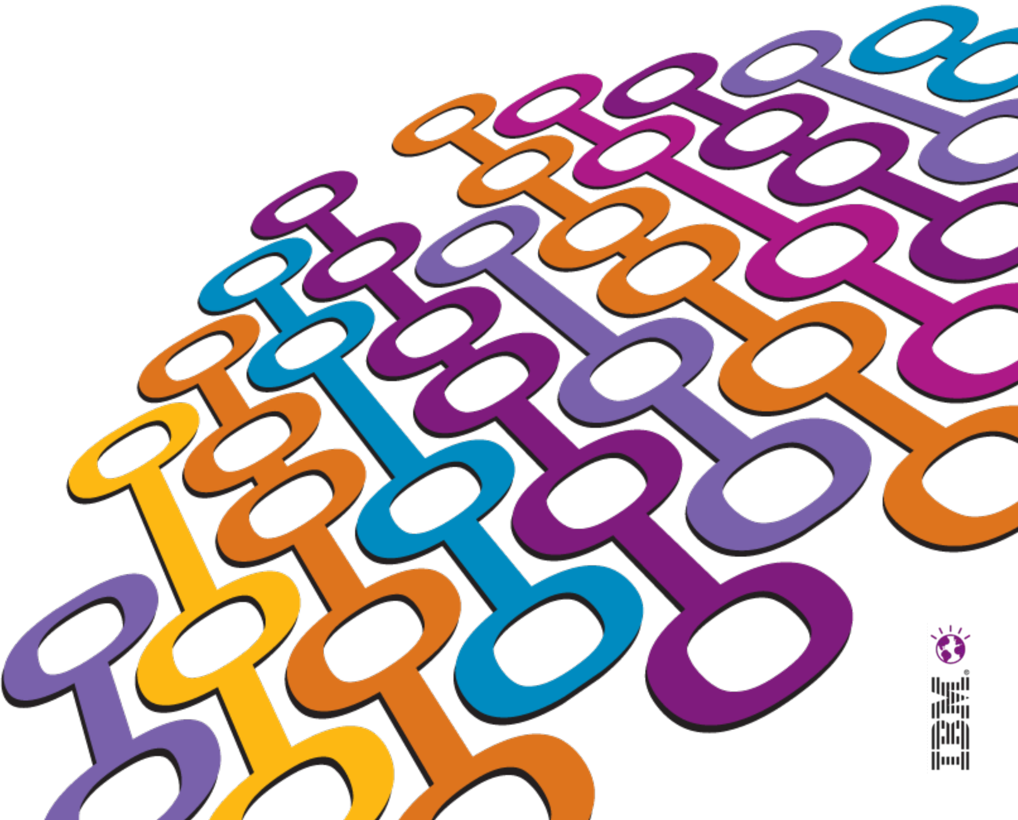
A.J. Aronoff, Connectivity Practice Director, Prolifics

T.Rob Wyatt, WebSphere C&I Product Manager, Security, IBM

Impact2012

The Premier Conference for Business and IT Leadership

Innovate. Transform. Grow.



Session Number #1869

Five MQ 7.1 Security Use Cases

0. Highlights of MQ 7.1 security

1. Enforce separation of production & non-prod MQ environments

2. Allow administrative gateway queue managers to provide more centralized MQ administration, change control and auditing

3. Leverage Message Broker Policy Enforcement Point nodes to provide security for message traffic that spans domain boundaries

4. Establish default levels of security. I.E. create MQ security Policies.

5. Secure message traffic from, to, and within MQ clusters

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.

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
Authorization 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
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

Today's True Tiny Tale of Terror

How to locate errant client application
MQSeries List Sent:Monday, April 30, 2012 11:58 AM
To:MQSERIES@LISTSERV.MEDUNIWIEN.AC.AT

Hello listers,

2-parter, here, extra credit will be given for answers to both of them.

I have a WMQ client application somewhere on one of 80+ distributed systems that is trying to connect to the same undefined channel every 5 minutes.

Using 'snoop' on the QMgr host, and using the timestamp in the AMQ9519 error message as a guide, I am still only able to narrow this down to about 20 of the hosts, and that is only if the timestamp on the AMQ9519 message is accurate to within 1 second.

1. Without impacting queue manager operations (no stop/start allowed), how can I locate the exact host and possibly even the correct application that is attempting the connections?

2. Once I locate the host, I believe I can use mqtrace to locate the application. Is there a better way to do this?

Thanks in advance for all assistance.

Five MQ 7.1 Security Use Cases

0. Highlights of MQ 7.1 security

1. Enforce separation of production & non-prod MQ environments

2. Allow administrative gateway queue managers to provide more centralized MQ administration, change control and auditing

3. Leverage Message Broker Policy Enforcement Point nodes to provide security for message traffic that spans domain boundaries

4. Establish default levels of security. I.E. create MQ security Policies.

5. Secure message traffic from, to, and within MQ clusters

Isolating Production & Non-Production Data

- Mixing Production & Non-Production data

  is more dangerous (career-wise) than mixing

  Matter & Anti-Matter

- Sometimes companies use a non-production machine

  as the high availability backup for a production machine.

  - What's the best way to *prevent an accident* where

    a non-production queue manager (or client),

    connects to a production queue manager?

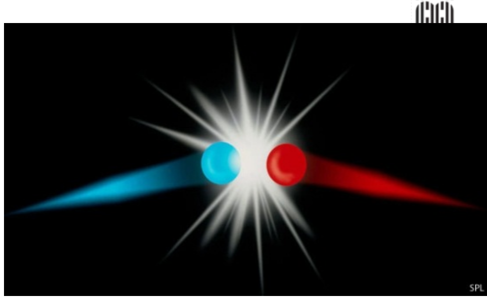
  - Some companies rely on firewall rules

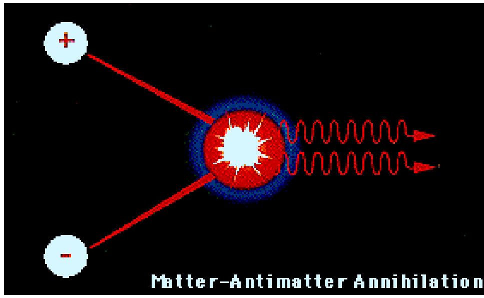
    - However, that is hard to do when a non-prod

      machine doubles as a production back-up.

- Happily, MQ 7.1 offers new security options
- This is much better than depending on luck.

Ray Charles: If it is wasn't for bad luck, I wouldn't have no luck at all





Security Features to Separate Prod & Non-Prod Data

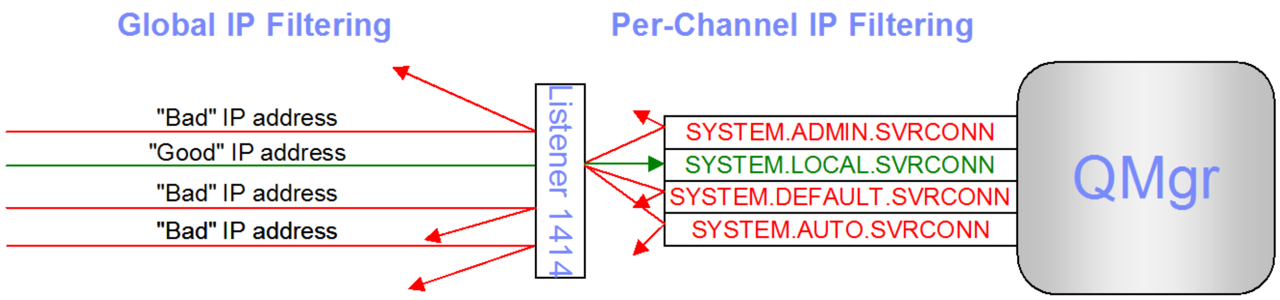
New Feature	*Benefits*	*Details*
IP address filtering	Allow or deny connections based on IP address	Blocks IP addresses at the listener and provides allow/deny functionality expressed as per-channel rules
User ID Mapping	Fine grained mapping of connection details to MCAUSER values	Allow or deny connections based on the ID presented in the connection request and map the ID to an MCAUSER
Remote Queue Manager Name	User Mappings can also use the name of the remote queue manager	Allow or Deny Channel Connections based on the name of the remote queue manager.
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

Isolate production & non-production with MQ 7.1

- Use IP Address filtering for isolation
  A. Create a generic wildcard rule that blocks all IP addresses by default
  B. A specific rule can then permit (whitelist) specific IP addresses / subnets
  - This is essentially a mini-firewall under the control of MQ administration
  - Note: Standard firewalls should also be used.
  - Note: Try to avoid production & non-production in the same sub-network
- Use Queue Manager Names for isolation
  - A key rule for a successful infrastructure is “Keep it Simple”
  - Simple MQ naming conventions help avoid ambiguity
  - Consider a new Queue Manager naming convention:
    - The first letter of a queue manager should identify the environment
      - P for Prod, D for Dev, T for Test, S for Staging …
    - A generic rule can then allow access to other queue managers from matching environments
  - Even better a specific receiver channel (FROMQMGR.TOQMGR) can be configured to only accept connections from QM1 from specific IP address(es)

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.



```
SET CHLAUTH('*') TYPE(BLOCKADDR) +
     ADDRLIST('192.168.1.*')
```

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

SET CHLAUTH('SYSTEM.LOCAL.SVRCONN') TYPE(ADDRESSMAP) +
     ADDRESS('127.0.0.*')  USERSRC(CHANNEL)
```

IP address filtering guidelines

- Keep in mind: *permitting specific IPs is better than forbidding specific IPs*. A permitted list (also known as whitelisting) says "here is an enumerated list of authorized requestors." A restricted list says "out of the practically infinite number of possible requestors, here are 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.

Dynamic mapping of MCAUSER

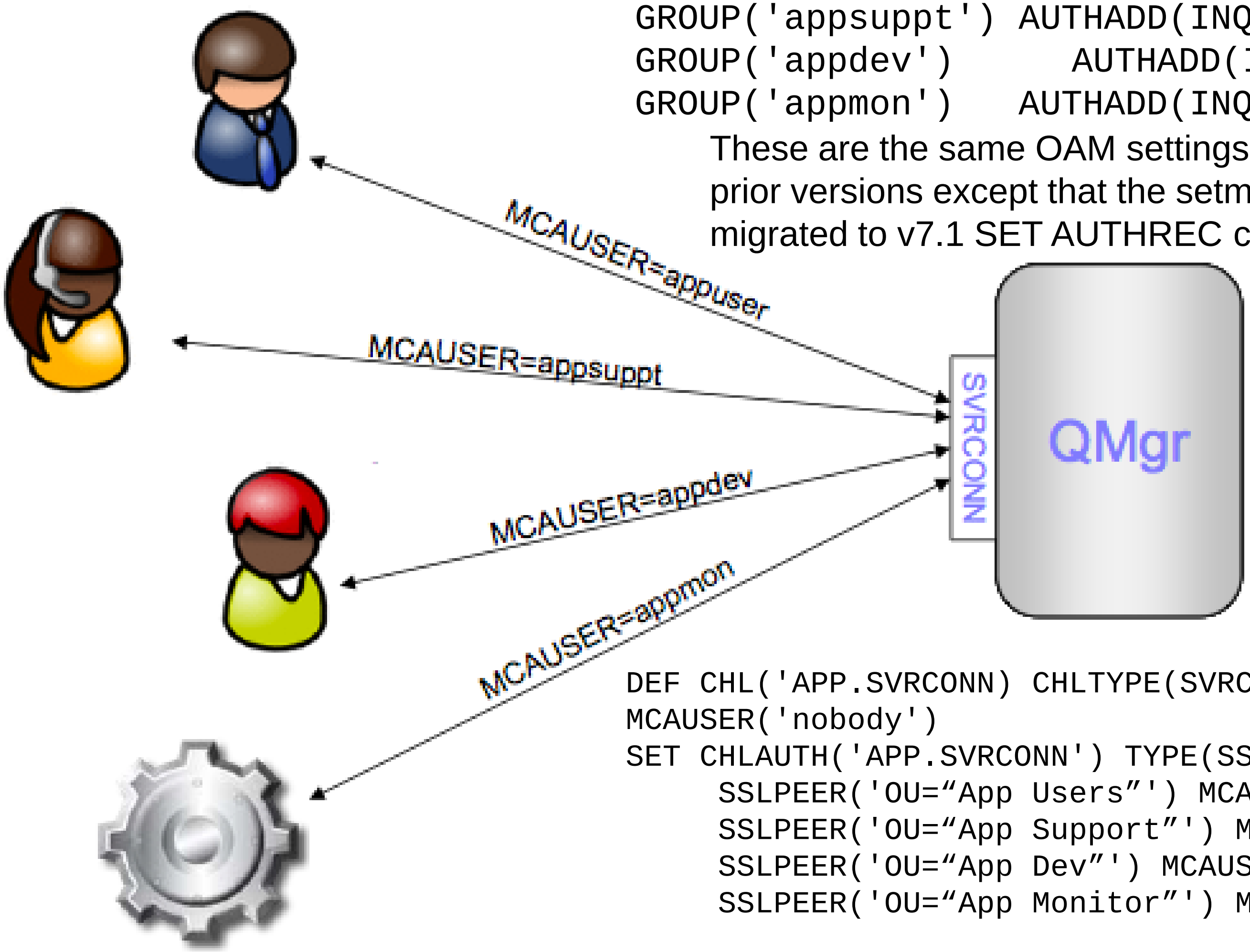
- The channel's MCAUSER determines the ID used for authorization, the same as in previous versions of WebSphere MQ.
- Authorization of remote entities is only as granular as the number of MCAUSER values.
- Prior to v7.1, administrators had a choice of either hard coding the value in the channel definition 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/sc83Ol
- 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 is 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

- An excellent article on MQ and SSL can be found at: http://bit.ly/MQSSLSecurity
- 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 (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 access rule. For example:

```
SET CHLAUTH(*) TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
SET CHLAUTH(*) TYPE(SSLPEERMAP) SSLPEER('CN=PQM7,O=IBM') USERSRC(CHANNEL)
SET CHLAUTH(*) TYPE(SSLPEERMAP) SSLPEER('CN=PQM7,OU=PROD, O=IBM') USERSRC(CHANNEL)
```

  - You can add the host name or IP Address to the SSL certificate
    - UNSTRUCTUREDNAME=AJSMACH1, UNSTRUCTUREDADDRESS=74.125/124.99
- **Note: It is a security risk to Clone Machines that have SSL Certificates**

Five MQ 7.1 Security Use Cases

0. Highlights of MQ 7.1 security

1. Enforce separation of production & non-prod MQ environments

2. Allow administrative gateway queue managers to provide more centralized MQ administration, change control and auditing

3. Leverage Message Broker Policy Enforcement Point nodes to provide security for message traffic that spans domain boundaries

4. Establish default levels of security. I.E. create MQ security Policies.

5. Secure message traffic from, to, and within MQ clusters

Centralized MQ Administration with Administrative Gateway Queue Managers

- You started the roll-out at 12:01 AM Sunday morning & requested the mqm password to make the change. The password has expired & everyone waits.
- Alternatively, imagine that all administrative changes can only be done via a secure administrative gateway queue manager.
  - The administrative gateway queue manager must be the most secure and well monitored system in the data center. No access without SSL Certificates, VPN,…
    - See next slide for details
  - The other queue managers (The ones that are being administered).
    - Provide a separate listener and Sender/Receiver channel that can only be accessed from the secure administrative gateway queue manager.
- The Queue managers that are being administered will publish all of their changes. (This feature has been available since 7.0.1).
- Centralized "Read Only" Queue Managers, can allow queue and channel statistical analysis, capacity analysis & error log analysis & a change log to be accessed, *without* allowing any access to the production queue manager

Administrative Gateway Queue Managers
Blocking mqm IDs except via the Admin gateway QM

- The CHLAUTH BLOCKUSER rules take effect after all other rules and exits have been processed & 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
- Blank user IDs resolve to the ID of the MCA which matches *MQADMIN.
- 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. E.g.
- SET CHLAUTH(*) TYPE(BLOCKUSER) USERLIST('nobody, *MQADMIN') SET CHLAUTH(QM.ADMIN.*) TYPE(BLOCKUSER) USERLIST('nobody')
- The first rule blocks administrative users and the MCAUSER 'nobody' (which prevents creating a user ID 'nobody' and putting it into an authorized group. Rules are hierarchical and the most specific one matches.
- The second rule provides a reduced blacklist for QM.ADMIN channels that allows administrators to use these. It is assumed here that some other CHLAUTH rule (e.g. SSLPEERMAP) has validated the QM.ADMIN channel

Administrative Gateway Queue Managers Depend on the Principal of Layered Defense

- First imagine a fort.
- Now add...
  - Perimeter buffer zone
  - Redundant fence
  - Razor wire
  - Flood lights
  - Spot lights
  - Moat
  - More!
- The idea is to place as many barriers as possible between your business assets and an attacker.

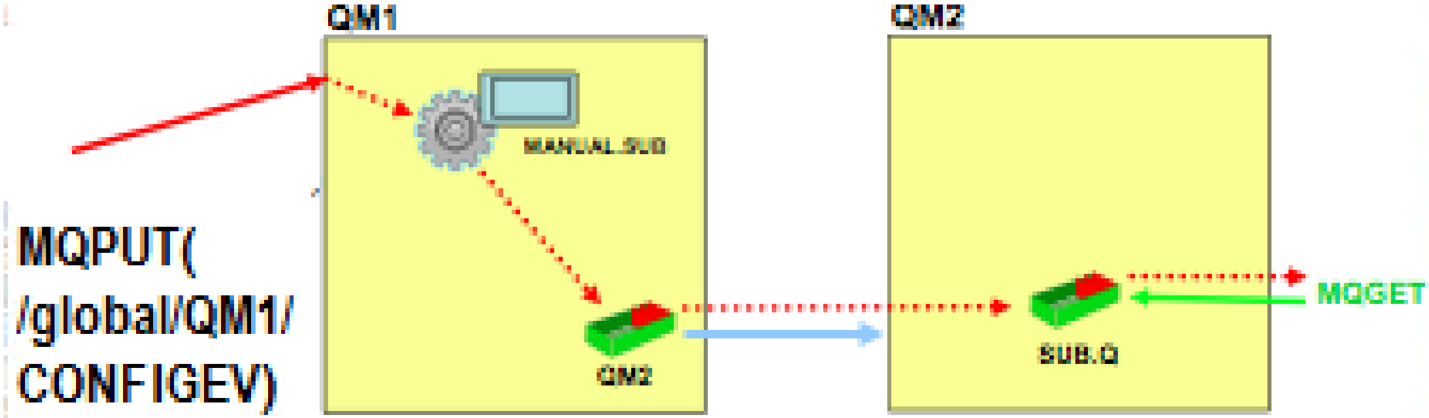
Administrative Gateway Queue Managers
Layered defense – WMQ style!

- MQ 7.1 Security policies should be used to ensure
  1. Only the gateway administrative security queue manager has privileged access to the other queue managers
  2. That SSL is required for every channel between the Administrative gateway and another queue manager.
- On the Administrative Queue Manager
  - Restrict access to initiation queues, XMit queues, command queue
  - Use non-standard listener port, bind listener to specific IP address
  - No application traffic is allowed via the administrative queue manager
  - Active intrusion detection through event and log monitoring
  - Channels can be bound to specific IP addresses
- Some of these mitigations (e.g. SSLPEER) set up barricades, some (e.g. removing unused CAs) minimize the “attack surface” while others (e.g. multiple listeners) provide resilience.
  - A blended approach is best.

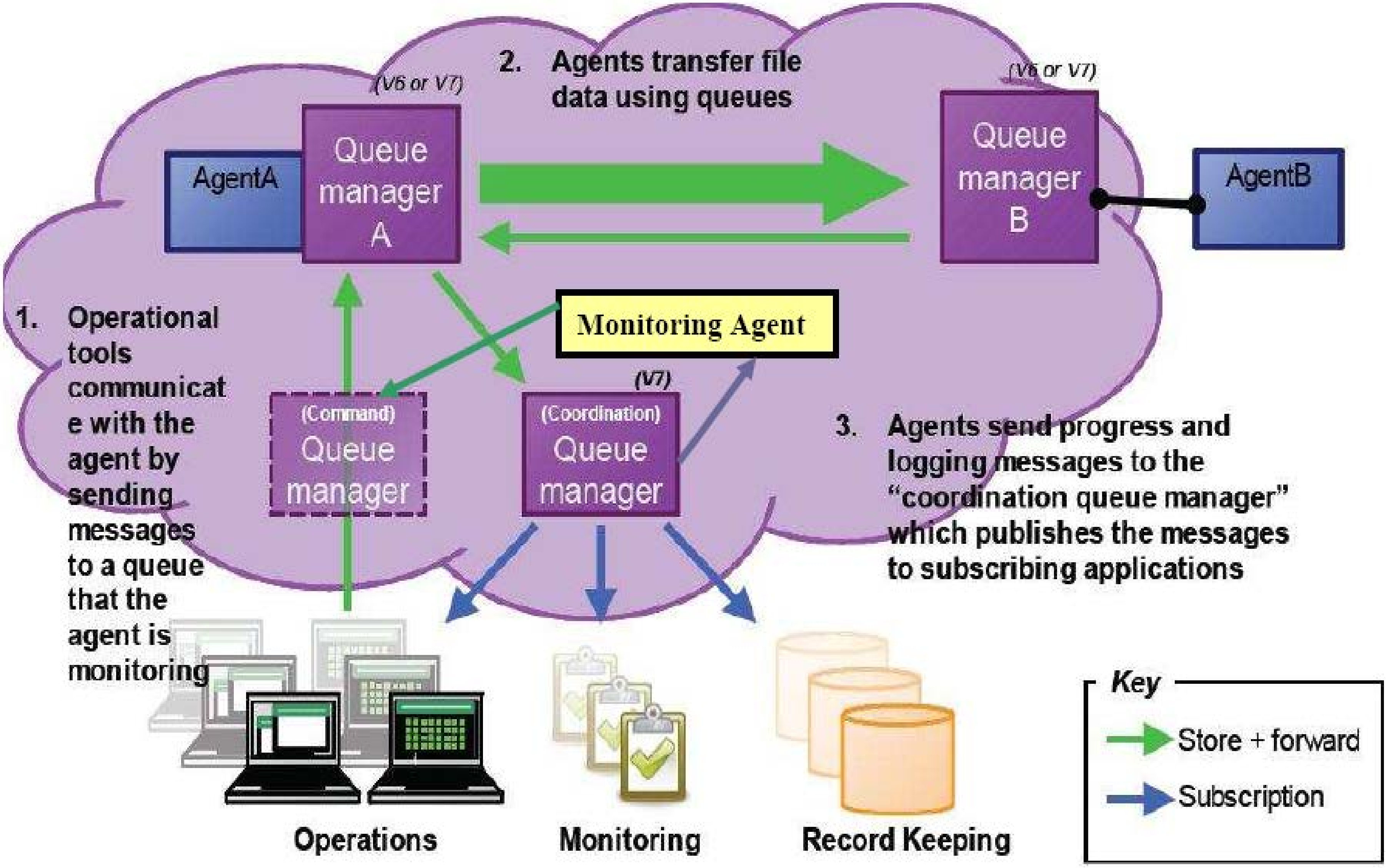
Centralized Security Monitoring Allows Publishing Security & Statistical info

- Enable Command Events and Configuration events
  - *Alter CMDEV(ENABLED) CONFIGEV(ENABLED)*
- Starting With MQ 7, Queues (including System Queues), can be replaced with topics.
  - Forward publications to a remote queue – QM1
  - DEF SUB(MANUAL.SUB) TOPICSTR('/global/QM1/CONFIGEV') DESTQMGR(QM2) DEST(SUB.Q)
  - Application gets message from SUB.Q
    - A SupportPac can monitor events

http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24029241&loc=en_US&cs=utf-8&lang=en



Read Only QMgrs for MQ FTE Pub/Sub Since all status information is published …



2. Agents transfer file data using queues

1. Operational tools communicate with the agent by sending messages to a queue that the agent is monitoring

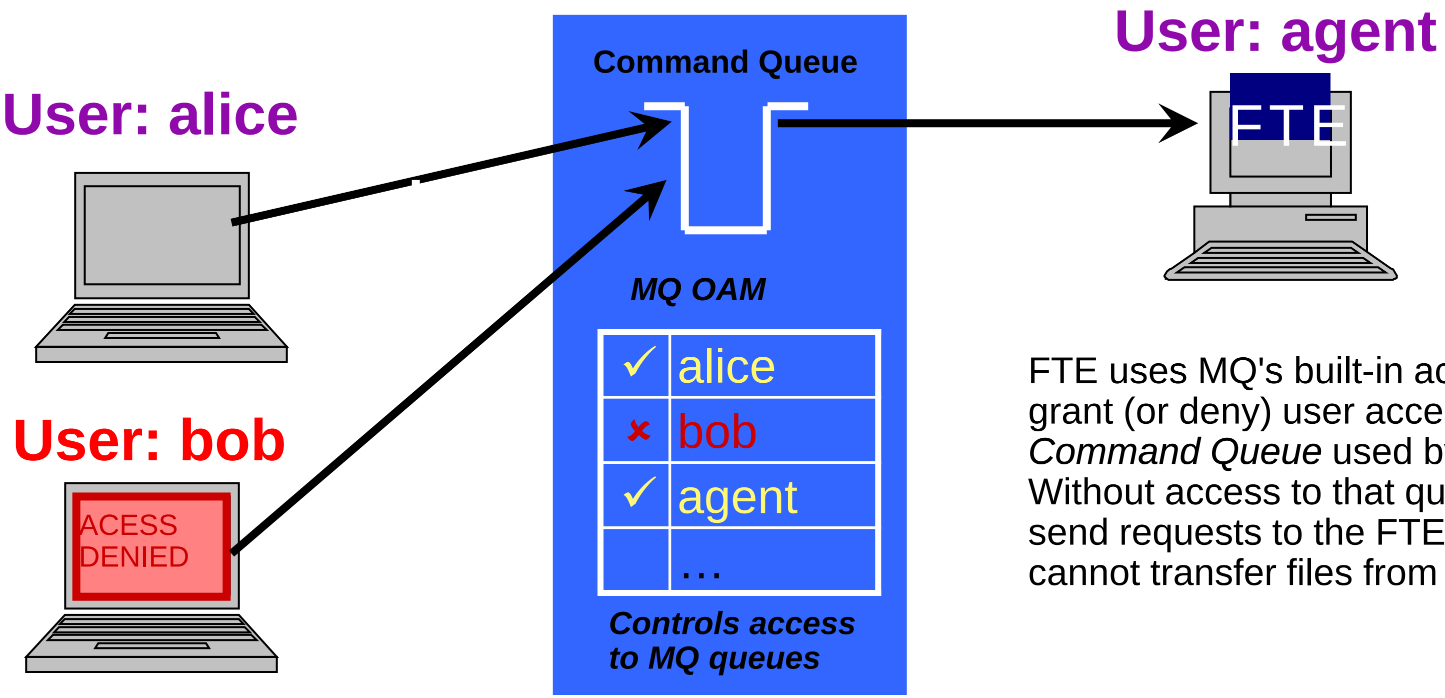
3. Agents send progress and logging messages to the "coordination queue manager" which publishes the messages to subscribing applications

Read Only QMgrs for MQ FTE Pub/Sub

- All MQ FTE information is published and organized by topic
- Inside the Coordinating Queue Manager permanent subscriptions can be manually defined and sent to a “Read only Queue Manager”
  - DEF SUB(MANUAL.SUB) TOPICSTR('/topic') DESTQMGR(READ.ONLY.QMNAME) DEST(SUB.Q)
- Access to the Coordinating Queue Manager can be limited to administrators.
- Only the Coordinating Queue Manager could send data messages to the READ.ONLY.QMNAME.
- Read access to the MQ FTE status messages on the Read Only Queue Manager could be easily (and securely) granted.
- Note: It is sometimes convenient to use the free Q Support Pac to search for particular strings in the status messages.
  - ./q –m QMNAME –i SUB.Q –h “a string”
  - Note: the –i option, means that the queue is being browsed

MQ FTE Security

- Provides the ability to easily configure finer grained access control to agent resources.
- This Depends on the user ID resolved by the channel.
  - MQ 7.1 user mappings make it easier to propagate IDs between QMgrs
- These IDs enable User and group based control of who can:
  - Transfer files to a particular agent
  - Transfer files from a particular agent
  - Perform operational and administrative agent functions



User: alice

User: bob

ACESS DENIED

Command Queue

MQ OAM

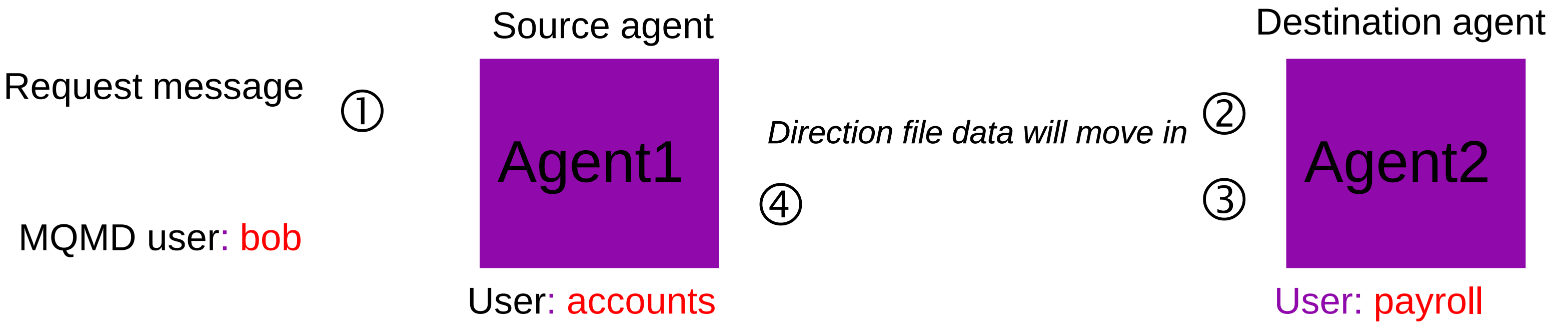
✓	alice
✗	bob
✓	agent
	…

Controls access to MQ queues

User: agent

FTE uses MQ's built-in access control to grant (or deny) user access to the *Command Queue* used by an agent. Without access to that queue, users cannot send requests to the FTE agent – and hence cannot transfer files from an agent.

Example of FTE authority checks that take place



Request message ①

MQMD user: bob

Source agent

Agent1

User: accounts

Direction file data will move in

④

②

③

Destination agent

Agent2

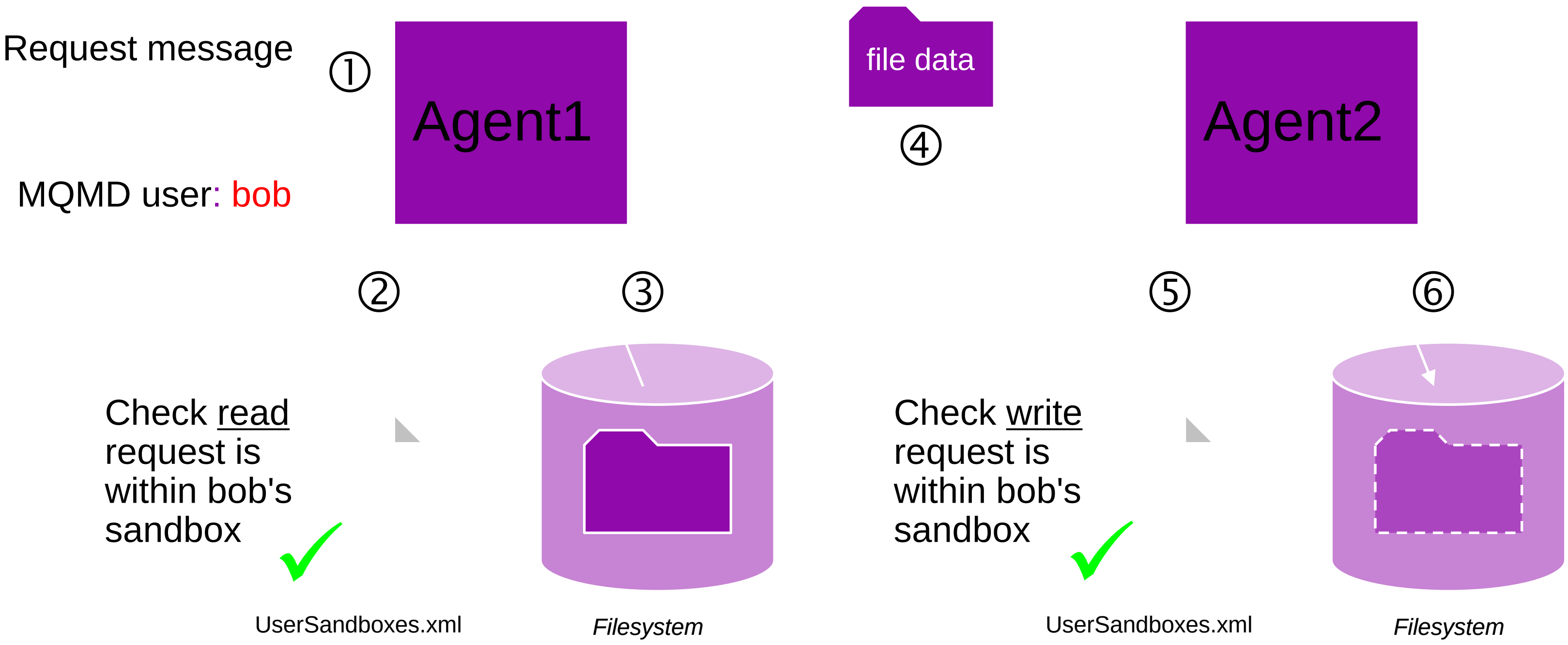
User: payroll

Checks that occur before the transfer starts:

1. Does 'bob' have 'transfer source' authority?
2. Does 'accounts' have 'agent source' authority?
3. *Does 'bob' have 'transfer destination' authority?*
4. Does 'payroll' have 'agent destination' authority?

Note that checks 1+4 happen at the source agent, whereas checks 2+3 occur at the destination agent

Overview of user sandboxes



Request message ①

MQMD user: bob

Agent1

file data

④

Agent2

② ③ ⑤ ⑥

Check read request is within bob's sandbox ✓

UserSandboxes.xml

Filesystem

Check write request is within bob's sandbox ✓

UserSandboxes.xml

Filesystem

- The checks at steps 2+5 must succeed otherwise the file transfer will not take place!

Security Best Practices: MQ Risk Review Group

- Systems keep growing and need periodic review
  - New applications are added (New fields have stronger security requirements)
    - Social Security Numbers, Credit Card information
  - New security standards are mandated (HIPPA, etc.)
- An MQ Risk Review Group should analyze new applications and scenarios
  - Does the new application require/warrant extra security?
  - A proactive approach to identifying and remediating risks
- This group should be a central point for MQ Risk analysis
  - Collect and analyze authorization event reports & incident reports
- Must meet on both a regularly scheduled and on an as needed basis
  - New scenarios that require analysis
  - New mandates, new regulations, new technology
  - Expansion (new locations)
  - Incident analysis
  - Mentoring/Knowledge transfer

MQ Security Practices *That Stand the Test of Time*

- Time is a key element of security strategies
  - Find/Fix vulnerabilities before:
    - an attacker can exploit them
    - an accident happens
  - Find vulnerabilities faster with centralized security monitoring
  - Deny administrative access with a gateway queue manager
    - Combining gateway QMs (layered defense) and centralied security monitoring increase the odds that an attack will be detected before it succeeds
  - Always separate Production (secure) & non-production (less secure)

Keep defenses up to date with an MQ Risk Review Group

Five MQ 7.1 Security Use Cases

0. Highlights of MQ 7.1 security

1. Enforce separation of production & non-prod MQ environments

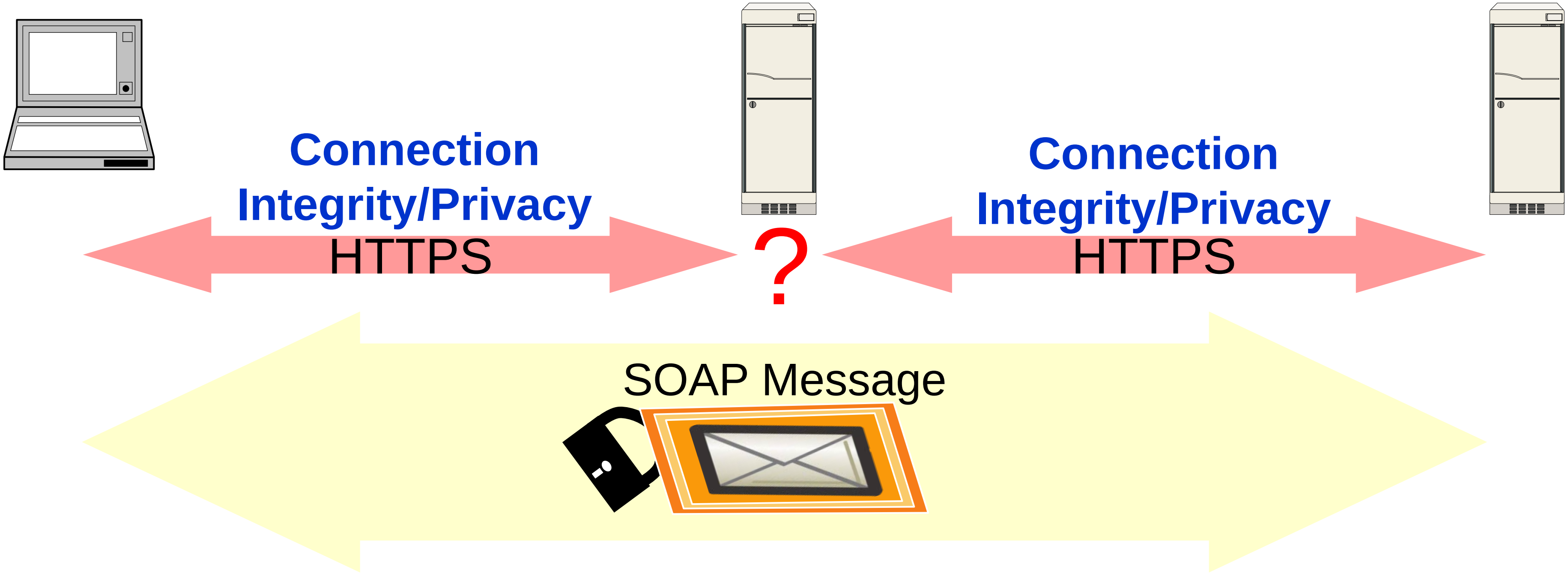
2. Allow administrative gateway queue managers to provide more centralized MQ administration, change control and auditing

3. Leverage Message Broker Policy Enforcement Point nodes to provide security for message traffic that spans domain boundaries

4. Establish default levels of security. I.E. create MQ security Policies.

5. Secure message traffic from, to, and within MQ clusters

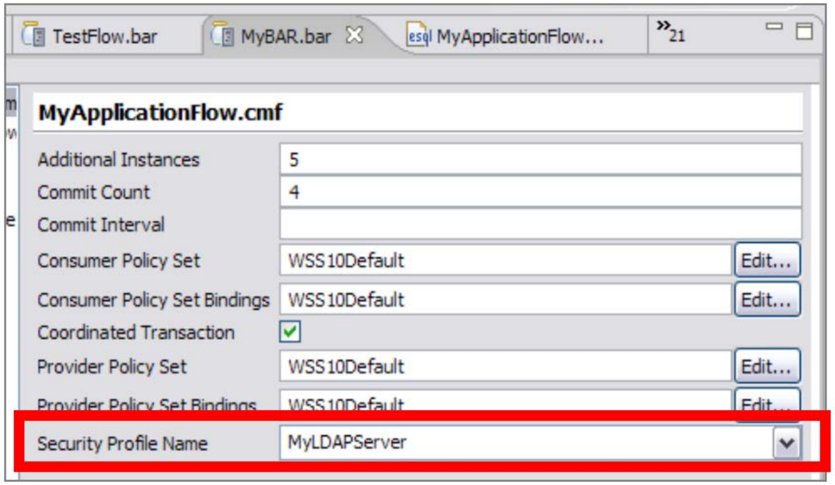
Mixing Protocols
Message-based Security : End-to-End Security



- Message-based security does not rely on secure transport
  - message itself is encrypted → message privacy
  - message itself is signed → message integrity
  - message contains user identity → proof of origin

Message Broker as a Policy Enforcement Point

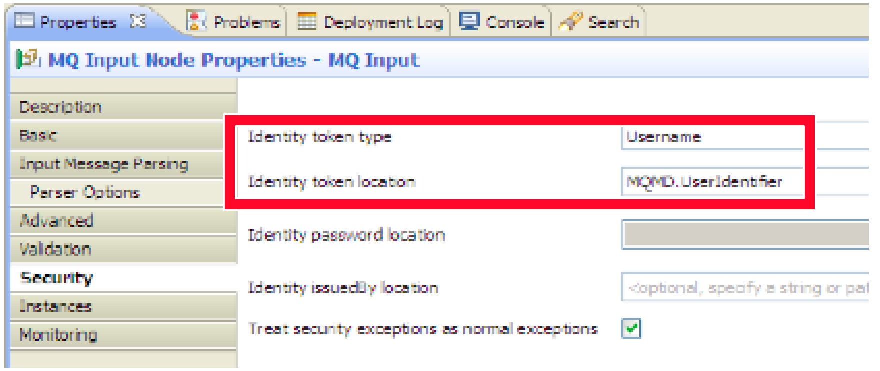
- ***Secure application identity, authentication and authorization***
- Application connectivity implies security domain changes
- Identity management, authentication, authorization & accounting mechanisms (AAA) are essential



MyApplicationFlow.cmf

Additional Instances: 5
Commit Count: 4
Commit Interval:
Consumer Policy Set: WSS10Default
Consumer Policy Set Bindings: WSS10Default
Coordinated Transaction: ☑
Provider Policy Set: WSS10Default
Provider Policy Set Bindings: WSS10Default
Security Profile Name: MyLDAPServer

- WMB supports many types of security tokens
  - Username, Username & Password, X509, SAML, Kerberos, LTPA, RACF
  - WMB supports many protocols & transports
    - Web Services, MQ, JMS, HTTP & HTTPS
- WMB performs role of Policy Enforcement Point (PEP) but does not authenticate by itself
  - Use with a Policy Decision Point (PDP) to provide a secure infrastructure
  - Ensures conformance to security policy
  - Many different PDP technologies supported
    - Lightweight Directory Access Protocol (LDAP)
      - Microsoft Active Directory, Open LDAP...
    - Tivoli Federated Identity Manager (TFIM)
    - WS-Trust



MQ Input Node Properties - MQ Input

Identity token type: Username
Identity token location: MQMD.UserIdentifier
Identity password location:
Identity issuedBy location: <optional, specify a string or pat
Treat security exceptions as normal exceptions: ☑

Security Drill Down

1st Layer Message Security

- Signature Validation/ Origin Authentication
- Message Level Decryption

2nd Layer Message Security

- Requestor Identification & Authentication & Mapping
- Element Level Decryption

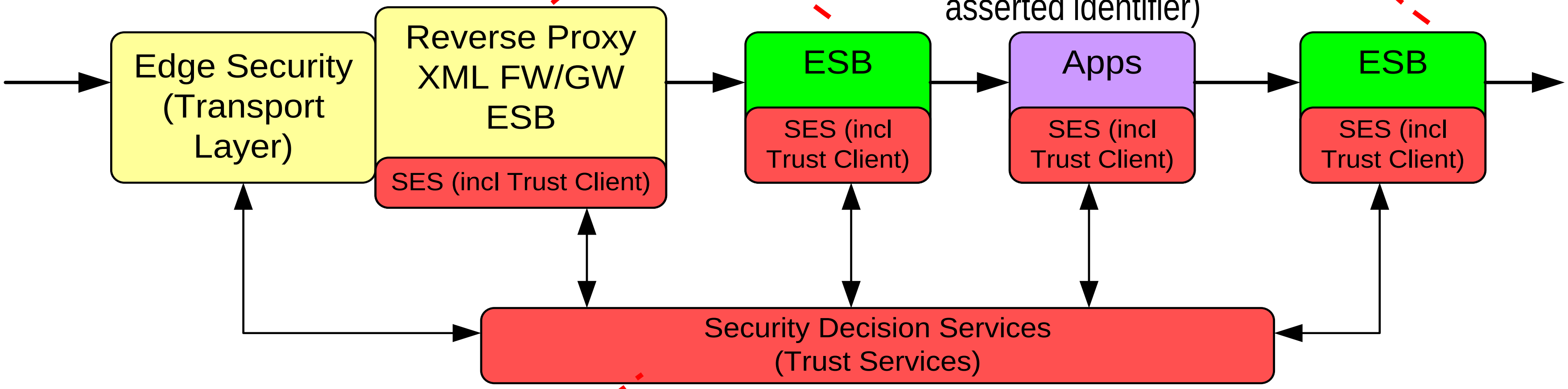
Nth Layer Message Security

- Requestor Identification & Authentication & Mapping
- Message Level Encryption

Transport Layer Security

- SSL/TLS Termination

Application Security (Authorization with ESB asserted identifier)



Edge Security (Transport Layer)

Reverse Proxy XML FW/GW ESB

SES (incl Trust Client)

ESB

SES (incl Trust Client)

Apps

SES (incl Trust Client)

ESB

SES (incl Trust Client)

Security Decision Services (Trust Services)

Security Policy

Security Token Service

Key Store, Management

Authorization

Five MQ 7.1 Security Use Cases

0. Highlights of MQ 7.1 security

1. Enforce separation of production & non-prod MQ environments

2. Allow administrative gateway queue managers to provide more centralized MQ administration, change control and auditing

3. Leverage Message Broker Policy Enforcement Point nodes to provide security for message traffic that spans domain boundaries

4. Establish default levels of security. I.E. create MQ security Policies

5. Secure message traffic from, to, and within MQ clusters

Establish Default Levels of Security: Blocking at the Listener

- Single list of IP address patterns
- 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 or userid 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

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

Establish MQ Security Policies
Channel Access Policy (1)

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

“We must make sure our system is completely locked down”

Establish MQ Security Policies
Channel Access Policy (2)

```
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)
```

“Our Business Partners must all connect using SSL, so we will map their access from the certificate DNs”

Establish MQ Security Policies
Channel Access Policy (3)

```
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)
```

“Our Administrators connect in using MQ Explorer, but don't use SSL. We will map their access by IP Address”

Establish MQ Security Policies
Channel Access Policy (4)

```
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(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"

Five MQ 7.1 Security Use Cases

0. Highlights of MQ 7.1 security

1. Enforce separation of production & non-prod MQ environments

2. Allow administrative gateway queue managers to provide more centralized MQ administration, change control and auditing

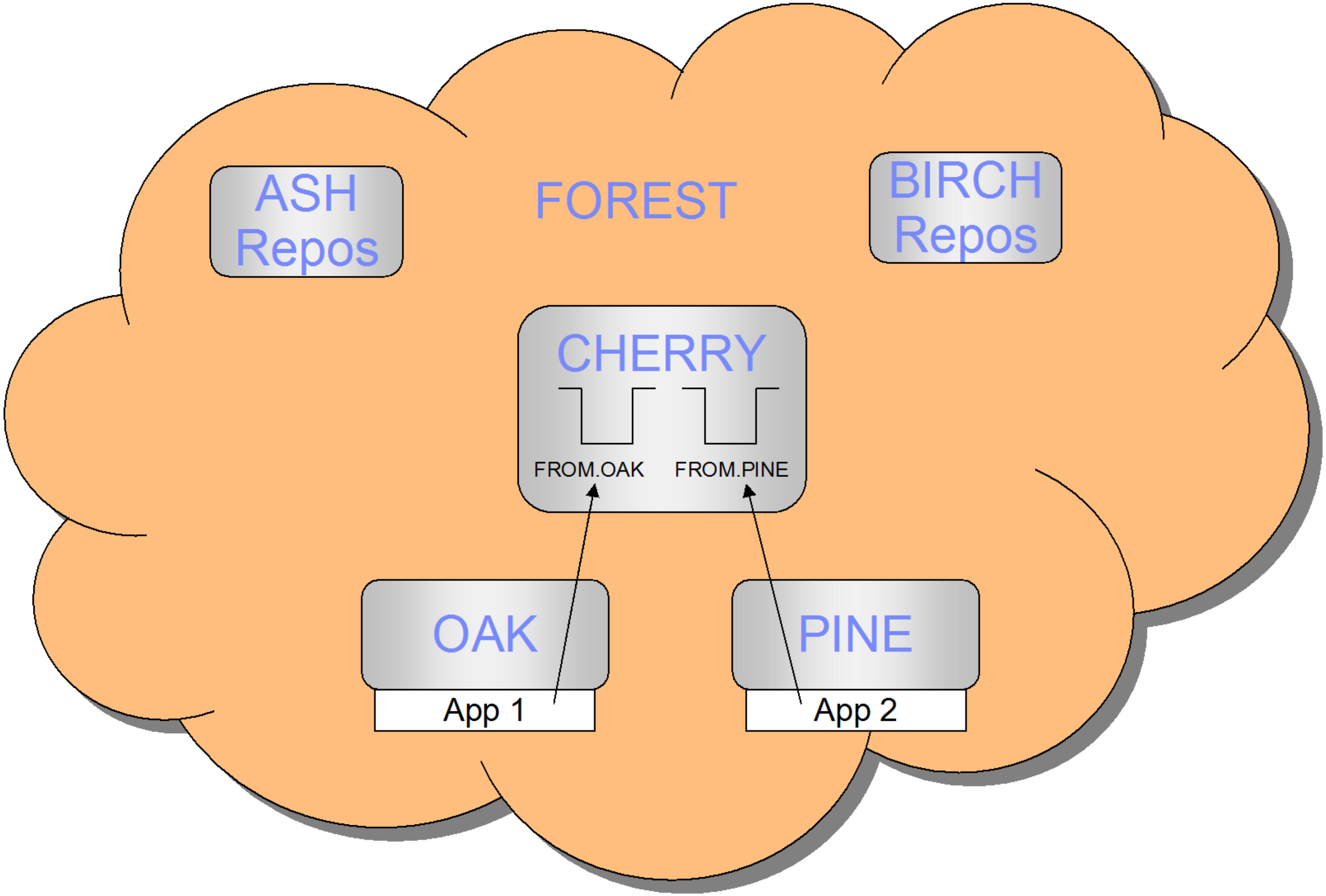
3. Leverage Message Broker Policy Enforcement Point nodes to provide security for message traffic that spans domain boundaries

4. Establish default levels of security. I.E. create MQ security Policies.

5. Secure message traffic from, to, and within MQ clusters

Secure message traffic from, to & within MQ clusters

- How can we secure access to the Cluster Below?



Secure message traffic from, to & within MQ clusters

- First Secure the Full Repositories (ASH & BIRCH)

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

SET CHLAUTH(CLUSTERNAME.ASH) TYPE(QMGRMAP) +
QMNAME(BIRCH)  MCAUSER(CLUSUSR) ADDRESS('9.30.50.84')
QMNAME(CHERRY) MCAUSER(CLUSUSR) ADDRESS('9.30.50.85')
QMNAME(OAK)    MCAUSER(CLUSUSR) ADDRESS('9.30.50.86')
QMNAME(PINE)   MCAUSER(CLUSUSR) ADDRESS('9.30.50.87')
```

- New Cluster Queue Managers will not join the cluster until the CHLAUTH rules for the full repositories are updated to allow them to join.
- Below are some generic CHLAUTH rules for non-repository cluster members. (Tighter rules next slide).

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

SET CHLAUTH(CLUSNAME.QMNAME) TYPE(ADDRESSMAP) ADDRESS('9.30.*') +
    USERSRC('see next slide')
```

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

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

Configuration on CHERRY QMgr

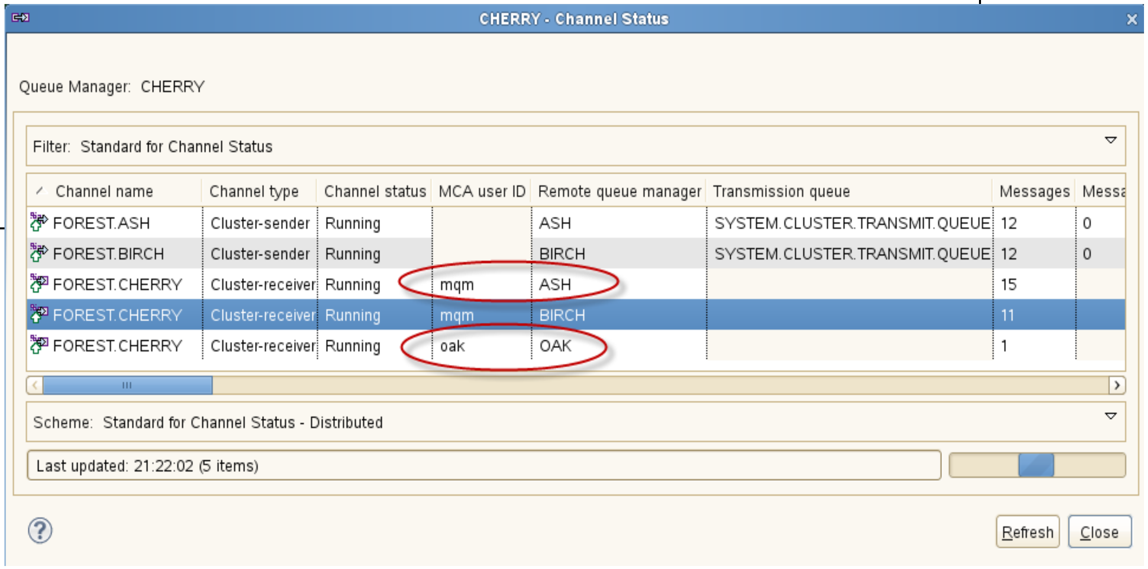
```
* MQSC Script

* 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:>
```



Queue Manager: CHERRY

Filter: Standard for Channel Status

Channel name	Channel type	Channel status	MCA user ID	Remote queue manager	Transmission queue	Messages	Messa
FOREST.ASH	Cluster-sender	Running		ASH	SYSTEM.CLUSTER.TRANSMIT.QUEUE	12	0
FOREST.BIRCH	Cluster-sender	Running		BIRCH	SYSTEM.CLUSTER.TRANSMIT.QUEUE	12	0
FOREST.CHERRY	Cluster-receiver	Running	mqm	ASH		15	
FOREST.CHERRY	Cluster-receiver	Running	mqm	BIRCH		11	
FOREST.CHERRY	Cluster-receiver	Running	oak	OAK		1	

Scheme: Standard for Channel Status - Distributed

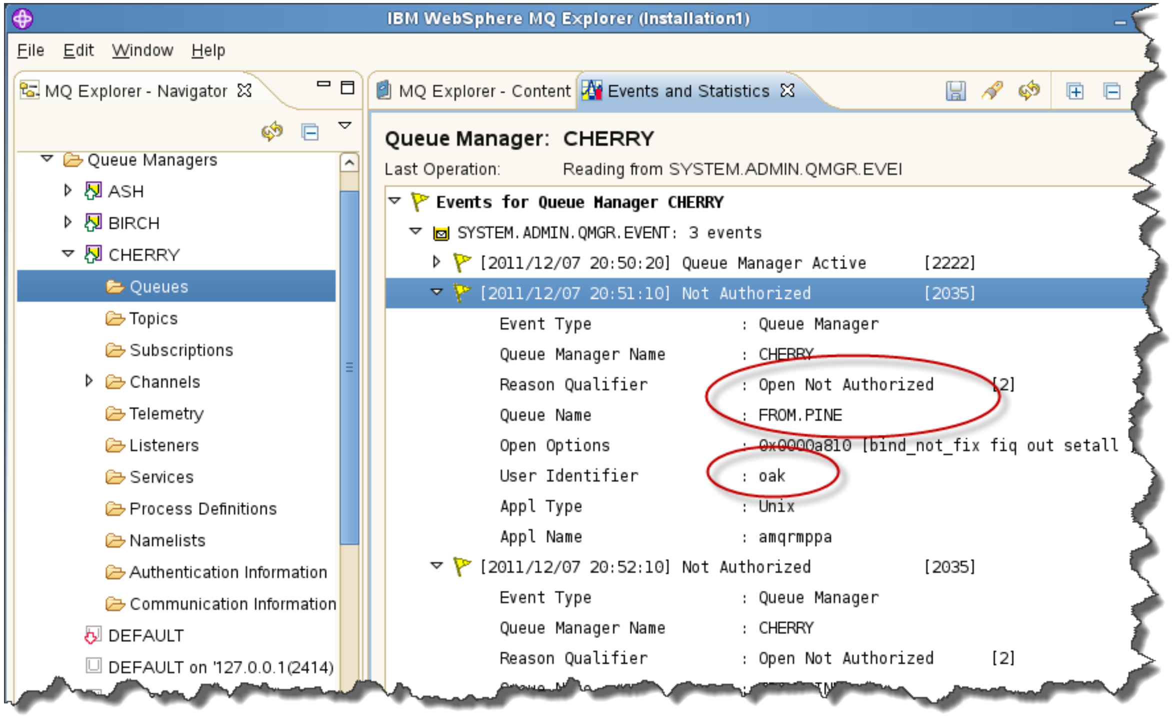
Last updated: 21:22:02 (5 items)

- 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



Queue Manager: CHERRY

Last Operation: Reading from SYSTEM.ADMIN.QMGR.EVEI

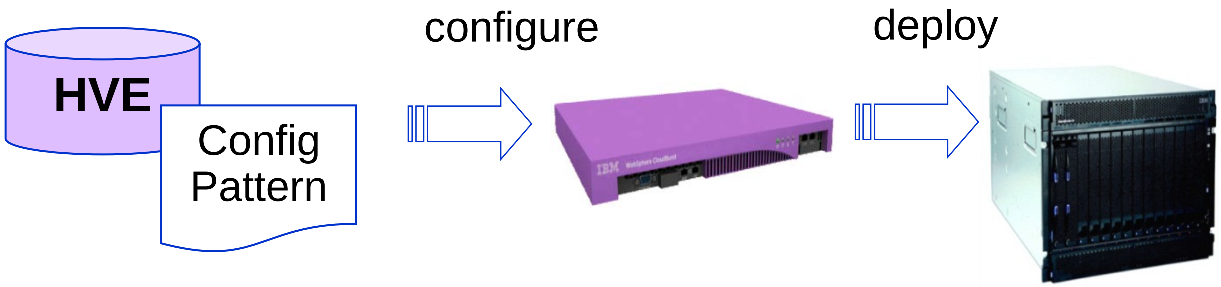
```
Events for Queue Manager CHERRY
  SYSTEM.ADMIN.QMGR.EVENT: 3 events
    [2011/12/07 20:50:20] Queue Manager Active      [2222]
    [2011/12/07 20:51:10] Not Authorized            [2035]
        Event Type              : Queue Manager
        Queue Manager Name      : CHERRY
        Reason Qualifier        : Open Not Authorized      [2]
        Queue Name              : FROM.PINE
        Open Options            : 0x0000a810 [bind_not_fix fiq out setall
        User Identifier         : oak
        Appl Type               : Unix
        Appl Name               : amqrmppa
    [2011/12/07 20:52:10] Not Authorized            [2035]
        Event Type              : Queue Manager
        Queue Manager Name      : CHERRY
        Reason Qualifier        : Open Not Authorized      [2]
```

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

MQ in the cloud: HyperVisor Editions

- HVE is pre-packaged image of MQ with an operating system
  - For easy configuration deployment into virtualised environments

- Pre-defined patterns for IBM WebSphere Workload Deployer



HVE

Config Pattern

configure

deploy

- On MQ Hypervisor, a script package can be used to add the queue manager to an existing WebSphere MQ cluster as a partial repository.
- All of the security techniques described in this presentation can be used with those scripts.

We love your Feedback!

- Don't forget to submit your Impact session and speaker feedback! Your feedback is very important to us, we use it to improve our conference for you next year.
- Go to impactsmartsite.com from your mobile device
- From the Impact 2012 Online Conference Guide:
  - Select Agenda
  - Navigate to the session you want to give feedback on
  - Select the session or speaker feedback links
  - Submit your feedback

Copyright and Trademarks

© IBM Corporation 2012. All Rights Reserved.

IBM, the IBM logo, ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Thank
You