MQ Security

A Holistic Approach

IBM CHAMPION

MQ Security Presentation Content

- Security Concepts
- Entities, Identities, and Authentication
- Identities, Resources, and Access Control
- Identities, Resources, and Auditing
- Resources and Privacy
- Summary

Security Concepts

The Big Picture

Entity → Identity → Operation → Resource

Security Concepts - I

- **Entity**
  - An abstract concept with “rights” to a resource
  - Person, Legal Entity (e.g. Corporation), Software (e.g. Application)
- **Identity**
  - An Entity may have multiple Identities
  - User IDs, X.509 Certificates, E-mail addresses
- **Resource**
  - Data, Commands/APIs, other Resources
  - Multiple types of access
    - Data access - CRUD (Create, Read, Update, Delete)
    - Command/API access (Execute)
    - Access

Security Concepts - II

Auditing

- Detection of penetration
  - Remember, **security always fails silently!**
- Evidence is both useful and is also a deterrent

Privacy

- An abstract concept covering all aspects of Security
- An end-to-end concept rather than a point in time or place

Layers ("Security in Depth")

- Multiple **independent** security mechanisms
- Reliability engineering; multiple independent breaches are less likely
- Attacks are not random & look for shared security dependencies

MQ Security – A Holistic Approach

Authentication

Connecting Identity to Entity

Authentication

Authentication Concepts

An Entity asserts an Identity

- Is the assertion valid?
- **All** subsequent steps **depend** upon this assertion!
- The asserted identity becomes a proxy for the actual entity or entities

Authentication mechanisms

- Possession of something physical
  - Physical Key, Fingerprint, Token Generator, etc.
- Knowledge of something Secret
  - User ID / Password
  - PKI KeyStore password (for x.509 certificates)

MQ Authentication – Local Connections I

Local Connections

- User ID associated with the client process (PID) by Operating System
- Local Operating System already authenticated process launch
- For MQ Connect (MQConn) API calls, this is "Server Bindings"
- A second layer of authentication may also be established (ConnAuth)

Connection Authentication settings

- Qmgr attribute ConnAuth non-blank (points to valid AuthInfo object)
- AuthInfo object AuthType equal to "IDPWOS" or "IDPWLDAP"
  - IDPWOS User ID/Password validated by Operating System
  - IDPWLDAP User ID/Password validated by LDAP

MQ Authentication – Local Connections II

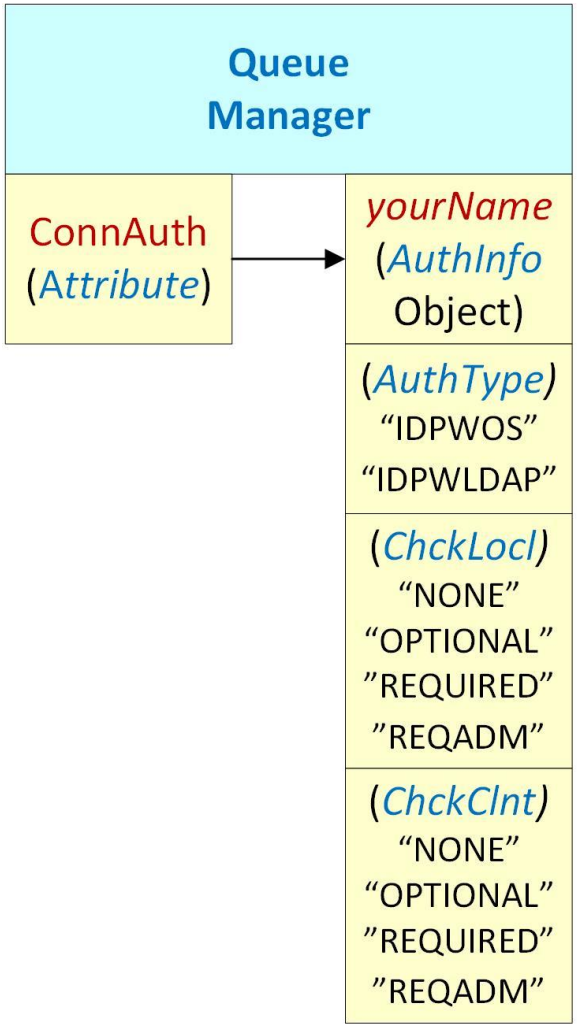
❖ Connection Authentication settings (continued)

- AuthInfo object ChckLocl equal to:
  - NONE — *No validation of User ID/Password*
  - OPTIONAL — *Validate only if User ID and Password provided*
  - REQUIRED — *User ID/Password required and validated*
  - REQADM — *User ID/Password required only for MQ admins*

❖ Security Exits

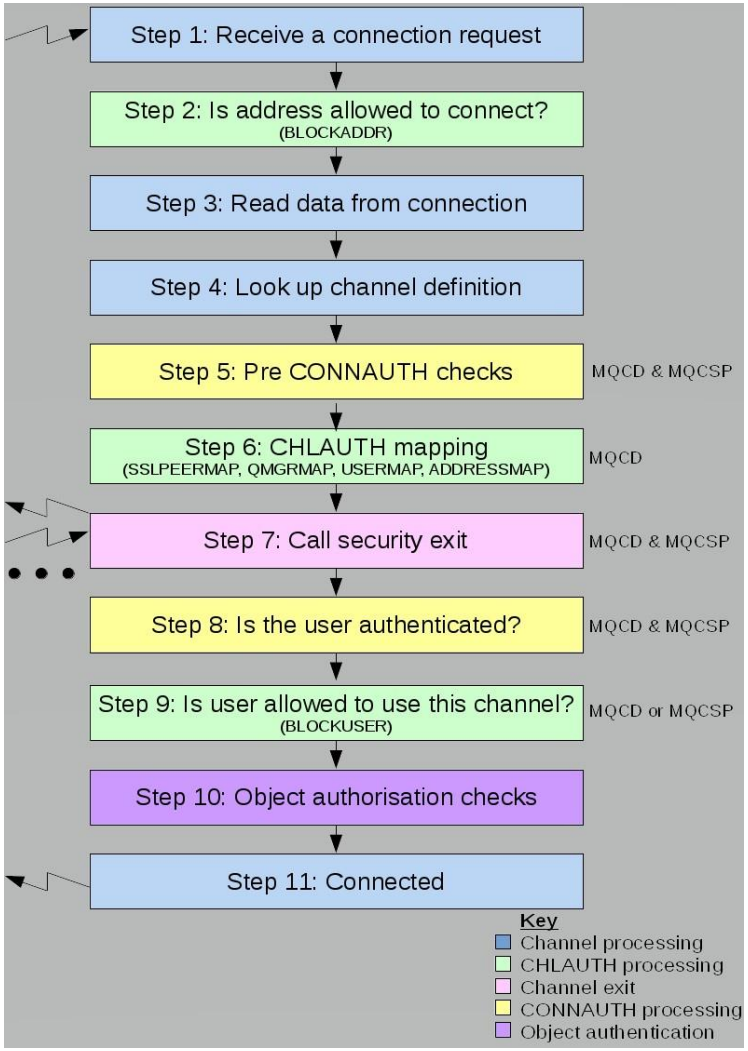
- No Channel Exits for local connections!
- **API Exits** may be implemented
  - MQConn, MQConnX (same exit)
  - Two exit functions for each MQI call (before and after MQI call)
- Develop Exits either in-house or acquire from a third party vendor
- IBM supplied sample API Exit (amqsaxe; source & executable)

MQ Authentication – ConnAuth Settings



- **ConnAuth Qmgr attribute (MQ v8.0)**
  - Qmgr Attribute set to non-blank
  - Qmgr Attribute points to AuthInfo object
- **AuthInfo Qmgr object**
  - AuthType
    - IDPWOS → Local OS validation
    - IDPWLDAP → LDAP validation
  - **ChckLocl**
    - Specify local host connection requirements
  - **ChckClnt**
    - Specify remote host connection requirements

MQ Authentication – Remote Connections I



- **Note the complicated interactions**
  - Channel Authorization rules may be passed different User ID values
- **This generated a number of APARs**
  - IT 12825 (ADOPTCTX=Y)
- **Configuration changes**
  - ADOPTCTX (Channel Authentication rules)
  - ChlAuthEarlyAdopt (qm.ini)
- **Unstable behavior across MQ versions**
  - T.Rob Wyatt documented 5 behaviors
- **Complicating factors**
  - Shared conversations (SHARECNV)
  - Connection Authentication (ADOPTCTX)
  - qm.ini (ChlAuthEarlyAdopt)
  - Channel (MCAUSER)

MQ Authentication – Remote Connections II

❖ Remote Connections (across a MQI channel)

- User ID asserted by the remote client
- Remote Operating System **may** be known, **may** be trusted
- Remote Operating System **may have** authenticated a User ID
- Remote Operating System **may** be passing the authenticated User ID

❖ SSL/TLS "Authentication"

- Possession of something physical (x.509 certificate)
- Possession of something secret (PKI KeyStore password)
- Identity asserted by x.509 certificate Distinguished Name (DN)
- SSL/TLS must be required for channel
- Certificate signer must be trusted
- Channel attribute SSLCAUTH must be set to "ENABLED"
- Channel attribute SSLPEER must be set to restrict allowed certificates

MQ Authentication – Remote Connections III

❖ Connection Authentication settings

- Qmgr attribute ConnAuth non-blank (points to valid AuthInfo object)
- AuthInfo object AuthType equal to “IDPWOS” or “IDPWLDAP”
  - IDPWOS User ID/Password validated by Operating System
  - IDPWLDAP User ID/Password validated by LDAP
- AuthInfo object ChckClnt equal to:
  - NONE *No validation of User ID/Password*
  - OPTIONAL *Validate only if User ID and Password provided*
  - REQUIRED *User ID/Password required and validated*
  - REQADM *User ID/Password required only for MQ admins*
- AuthInfo object AdoptCtx equal to:
  - YES *User ID presented for authentication is used*
  - NO *User ID asserted (but not authenticated) is used*

MQ Authentication – Remote Connections IV

❖ Channel Authentication settings

- Not authentication at all, more like firewall rules
  - Define excluded IP addresses and User IDs
  - Define allowed IP addresses and User IDs
  - Map incoming credentials to a different User ID
  - No validation performed!
- Provides some protection but provides no authentication
- May map presented User ID to another User ID

❖ Configuration file (qm.ini) settings

- ChlAuthEarlyAdopt - Use ConnAuth ID for ChlAuth rules

❖ Channel settings

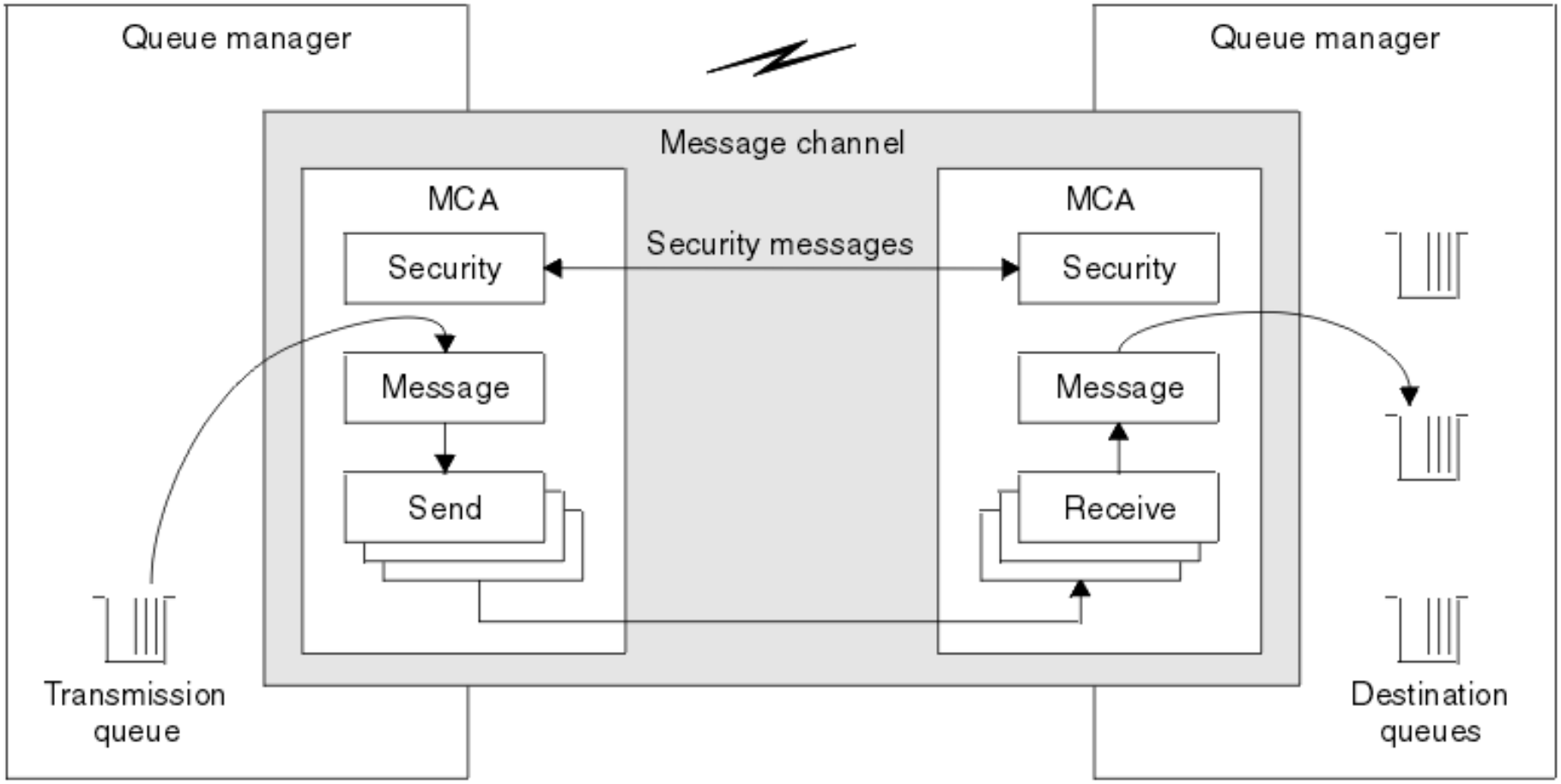
- MCAUser - Defines Message Channel Agent (MCA) authority

MQ Authentication – Remote Connections V

❖ Security Related Exits

- Supported Channel Exits
  - Security Exits (MCA Client & MCA Server; Message & MQI channels)
  - Send/Receive Exits (Source & Destination; Message & MQI channels)
  - Message Exits (Source & Destination; Message channels only)
  - API Exits (MQConn, MQConnX)
- IBM Provided **Channel Exit** programs
  - Security Support Provider Interface (SSPI) Exit (Windows only)
    - IBM provided source code & executable
    - Available for both MQ Client and MQ Server environments
  - CSQCAPX – API Crossing sample assembler program (z/OS only)
- Develop Exits either in-house or acquire from a third party vendor
  - BlockIP / BlockIP2 (introduced in 2002)
    - Jorgen Pedersen, Michael Dag, Sid Young, Neil Casey, et al
  - MQAUSX and other products from Capitalware
    - Roger Lacroix

MQ Authentication – Channel Exits

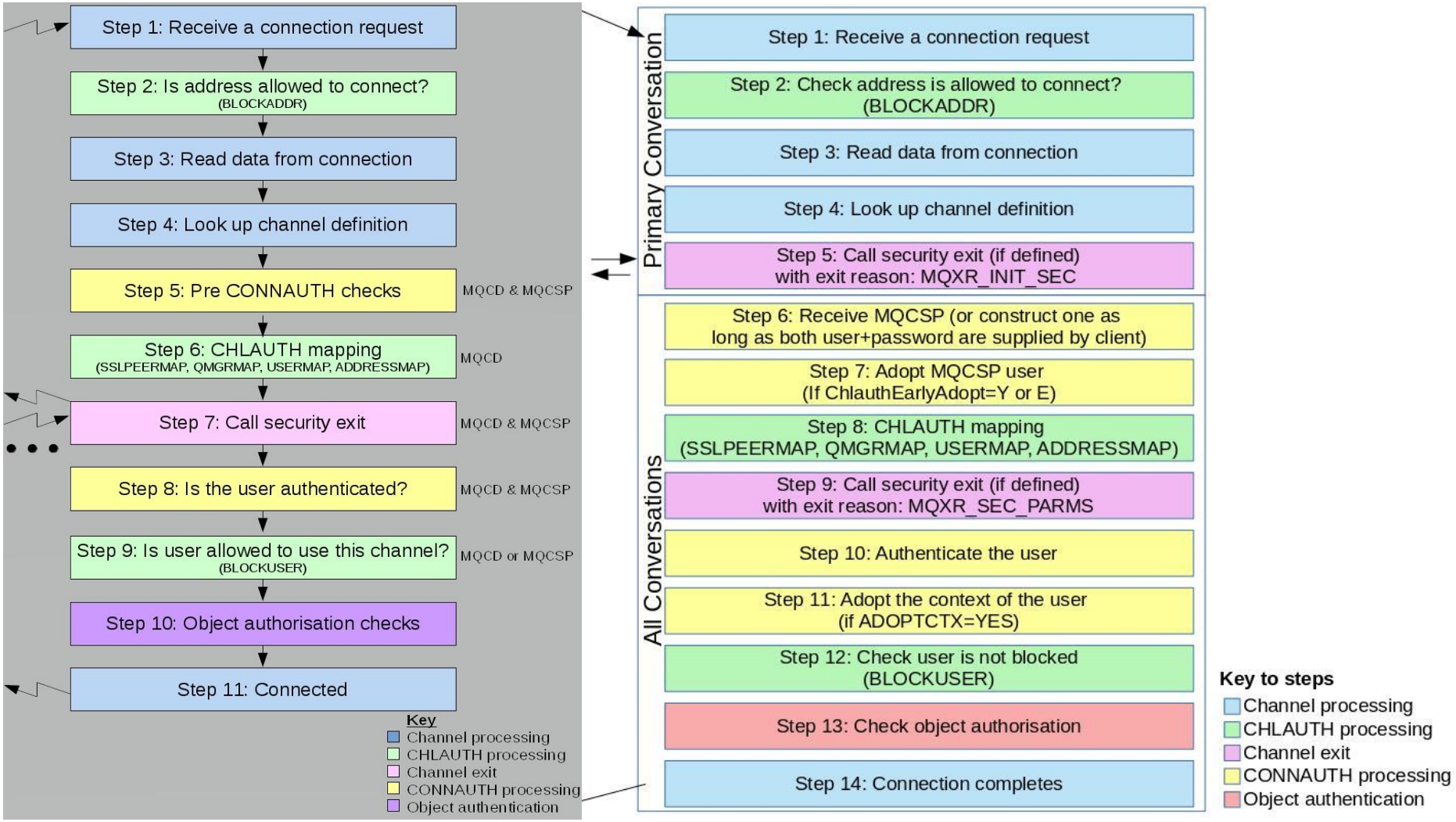


MQ Authentication – Remote Connections VI

❖ Interplay between security processes

- Interaction between Channel Authentication, Connection Authentication, and MQ Exits has historically been "brittle"
- Numerous APARs addressing the interplay between these features
- Originally, only an article on developerWorks by Mark Wilson
- T.Rob Wyatt (IBM Champion) has produced an entire presentation on the interplay between ConnAuth and ChlAuth
- IBM did not "officially" specify the security architecture & precedence between all of these features until August 2018 and that was in an APAR (IT 25839)

IBM APAR IT 25839 (Latest and Last?)



ConnAuth & ChlAuth Notes I

N

O

T

E

S

Mark Wilson – IBM Hursley Laboratory - developerWorks
The interaction of CHLAUTH and CONNAUTH in IBM MQ
https://www.ibm.com/developerworks/community/blogs/messaging/entry/The_interaction_of_CHLAUTH_and_CONNAUTH_in_IBM_MQ?lang=en_us

T.Rob Wyatt
IBM MQ CONNAUTH/CHLAUTH Doesn't Work Like You Think it Does
https://www.slideshare.net/tdotrob/ibm-mq-connauthchlauth-doesnt-work-like-you-think-it-does-and-if-you-arent-careful-may-not-work-at-all

Selected APARs
IT 08408, IT 12825, IT 17824, IT 18052, IT 20275, IT 25591, IT 25839
PI 41329, PI 61543, PI 63228, PI 97781, PI 98314

APAR IT 25839
https://www-01.ibm.com/support/docview.wss?uid=ibm10725873

ConnAuth & ChlAuth Notes II

N

O

T

E

S

MQ “Feature” Interactions
==========

- ✓ **Shared Conversations** (SHARECNV) - Introduced in MQ v7
- ✓ **Channel Authentication** (CHLAUTH) – Introduced in MQ v 7.1
- ✓ **Connection Authentication** (CONNAUTH) – Introduced in MQ v8
- ✓ **ChlAuthEarlyAdopt** (qm.ini parameter) – Introduced from APAR (IT 12825) in MQ v8.0.0.5

Available Security Exits
==========

- ✓ BlockIP2 (http://www.mrmq.dk/joomlaEN/en/)
- ✓ MQAUSX (https://www.capitalware.com/mqausx_overview.html)

Authentication Summary - I

One Entity

- The “Client”

Multiple Possible Identities

- Client **Application** User ID
- Client User ID/Password (**ConnAuth**)
- Client **x.509 Certificate** Common Name
- Channel Auth mapping (**ChlAuth**)
- **Security Exit** assigned User ID
- Channel assigned User ID (**MCAUser**)

Authentication Summary - II

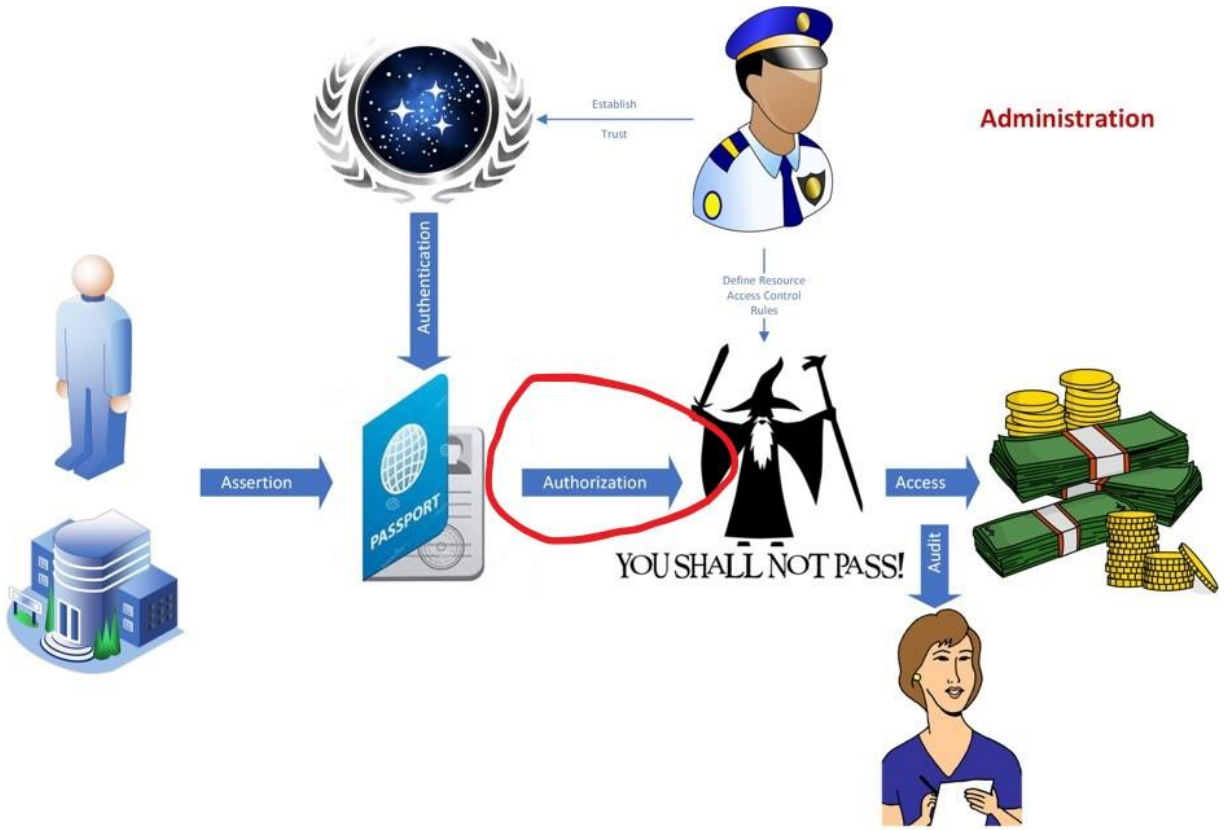
Authentication Factors	Security Mechanisms
Firewall	Weak; no authentication
Channel Authentication	Weak; no authentication
Connection Authentication	Strong; Possession of secret
SSL/TLS	Strong; Possession of object (certificate) & secret (password)
Channel (MCAUser)	Weak; no authentication
Security Exits	Varies depending upon Exit behavior

MQ Security – A Holistic Approach

Access Control

Allowing Identities access to resources

Access Control Authorization



Entity	Identity	Operation	Resource
Person	*Name*	*Administration*	*Queue Manager*
Corporation	*Email Address*	*Operation*	*Channel*
Server	*URL*	*I/O*	*Queue*
etc.	*etc.*	*Support*	*MQ Object*

MQ Object Authority Manager (OAM) I

Grant scope granularity

- Granted to individual User IDs (Be sure Qmgr is configured for this!)
- Granted to groups

Permission granularity

- Minimum permissions (e.g. Browse)
- Average permissions (e.g. Browse, Get, Inq, Pub, Put, Sub)
- Blanket permissions (AllMQI)

Object granularity

- Object name wildcards (e.g. HLQ.**)

Operational standardization

- Grant by hand vs Grant by script (preferred)
- Separate MQSC/Grant definitions vs Combined definitions (preferred)
- Configuration scripts managed in repository (preferred)
- Incremental grants (+privileges) vs Full grants (-all then +privileges)

MQ Object Authority Manager (OAM) II

Maintenance of permissions

- Processes in place to trigger removal of permissions
  - External to MQ Admins (e.g. LDAP group membership)
  - MQ Admin responsibility
- Processes in place to trigger removal of permissions
  - If not, assumption is a history of perfection (Not self-correcting)
- Processes in place to audit permissions
  - Granting by Principal dramatically increases challenge
  - Granting by specific object name dramatically increases challenge
  - In most instances, some kind of automation is required

MQ Security – A Holistic Approach

Auditing

Reactive rather than proactive security

MQ Security Monitoring & Auditing

Entity	Identity	Operation	Resource
Person	*Name*	*Administration*	*Queue Manager*
Corporation	*Email Address*	*Operation*	*Channel*
Server	*URL*	*I/O*	*Queue*
etc.	*etc.*	*Support*	*MQ Object*

Security Failures

❖ Two types of security failures

- False Positives (Too much security prevents legitimate access)
  - Authorized Users make a lot of noise when this happens
- False Negatives (Breech!)
  - These are silent failures!

❖ Audits are required even with perfect security

- Required to determine inappropriate use of authorized access
- Required as evidence

❖ Audits are a self-correction mechanism

- Audit permissions for unauthorized grants
- Audit access for unusual patterns (easy to state but hard to do)

MQ Security Monitoring

❖ MQ Error Logs

- Verbose log; what to look for?
- Error messages not easily selected
- Security error messages evolve across releases
- Beta feature: Logs in JSON format
  - Environment Variable AMQ_ADDITIONAL_JSON_LOG=1

❖ MQ Error Logs in JSON Format

- Beta level feature; Additional “.JSON” Log files created
- Set Environment Variable “AMQ_ADDITIONAL_JSON_LOG” = 1

Error Log Messages – Distributed

NOTES

- AMQ4036 (Not authorized; MQRC 2035)
- AMQ4079 (Channel closed by security exit)
- AMQ8135 (Not authorized; MQRC 2035)
- AMQ8242 (Invalid CipherSpec)
- AMQ8604 (Not authorized; Trigger Monitor)

Error Log Messages – Windows

N

O

T

E

S

- AMQ8063 (Not authorized; Command)
- AMQ8064 (Not authorized; Start trusted Application)
- AMQ8072 (Not authorized; Administer channels)
- AMQ8073 (Not authorized; SID resolution failed)
- AMQ8076 (Not authorized; OAM not supplied with SID)
- AMQ8081 (Not authorized; Administer Qmgr)
- AMQ8082 (Not authorized; Administer Clusters)

MQ Security Auditing

Queue Manager Events

- Enable Queue Manager Events
  - alter qmgr CONFIGEV(ENABLED)
  - alter qmgr AUTHOREV(ENABLED)
- Event messages written to:
  - SYSTEM.ADMIN.CONFIG.EVENT (Configuration events)
  - SYSTEM.ADMIN.QMGR.EVENT (Security events)

IBM Supplied Event Queue Monitor Sample Program

- amqsevt -m *QmgrName* -q *eventQueueName* -b
  - “-b” browses rather than destructively gets messages
- SupportPac MH05 - xmqdspev (Oliver Fisse)

MQ Security – A Holistic Approach

Layers & Privacy

Security in depth

Security Layers

- **If security were perfect,**
  - We wouldn't need monitoring
  - We wouldn't need layers of security
  - We would still need auditing to detect invalid use of authorized access
- **Security in depth**
  - Multiple **independent** layers harder to penetrate than a single layer
    - Network security (Firewalls)
    - Operating System security (Administration)
    - File System security (Operating System)
    - Link level security (SSL/TLS)
    - MQ Security

End-to-End Security

- **If security were perfect,**
  - We wouldn't be talking about privacy
- **Data in Motion**
  - Messages transmitted across a Message channel
  - Messages read or written across a MQI channel
- **Data at Rest**
  - Messages in a Queue
  - Messages in the Log
- **x.509 Certificate Skills**
  - No longer an optional skill set

Advanced Message Security (AMS)

❖ AMS available in MQ Advanced license

- Encrypts data at rest
  - Transparent to the Application
  - Message encrypted by intercepting the “Put” API call
  - Logged message is therefore also encrypted
- New (MQ v9) Quality of Protection: Confidentiality
  - Performance dramatically improved
  - Prediction, this will become normal rather than exotic
- Based upon Public Key Infrastructure (PKI) x.509 certificates

MQ Security – A Holistic Approach

Summary

Totum maior summa partum

Looking at the Whole: Things to Consider

MQ Clusters

- A MQ Cluster establishes a “zone of trust”
  - Any Cluster Queue Manager may connect to any other
    - Additional security may be needed to prevent unwanted connections
  - Additional monitoring & auditing needed to detect intrusions
- Smaller clusters provide “Bulkheading” to limit scope of penetration

Command Server

- MQ’s most dangerous feature (from a security point of view)
  - Enables a Single Point of Control (SPOC)
  - For administrators, only required for remote administration
  - Tools may depend upon it

More Things to Consider

❖ Triggering

- Trigger Monitors launch processes
  - Potentially complete access to everything available to it's User ID
    - Run Trigger Monitors under the most restrictive User ID possible
  - Additional monitoring & auditing may be required

❖ Channel Auto Definition

- Normally turned off
- But isn't this what Cluster do?
- Again, if used, additional monitoring & auditing will be needed

❖ "Trusted" Applications

- IBM Integration Bus (IIB) / App Connect Enterprise (ACE)
  - Consider running under their own User ID
  - If run as MQ administrator, limit Queue Manager scope

A Comprehensive Security Strategy

❖ Consistent Authentication strategy without gaps

- Both Local and Remote connections
- Both Message and MQI channels

❖ Consistent Access Control

- Control access by Principal or by Group
- Control access across multiple computing Platforms
  - IBM i, UNIX, Windows, z/OS
- Control access across through multiple security software programs
  - Active Directory, LDAP
  - mainframe (e.g. RACF), MQ OAM, UNIX OS, Windows OS

❖ Auditing Strategy

- MQ Error Log
- Security Events

Questions & Answers

Presenter

- Glen Brumbaugh
  - Glen.Brumbaugh@TxMQ.com
- Computer Science Background
  - Lecturer in Computer Science, University of California, Berkeley
  - Professorial Lecturer in Information Systems, Golden Gate University, San Francisco
- WebSphere MQ Background (25 years plus)
  - IBM Business Enterprise Solutions Team (BEST)
    - Initial support for MQSeries v1.0
    - Trained and mentored by Hursley MQSeries staff
  - IBM U.S. Messaging Solutions Lead, GTS
  - Platforms Supported
    - MVS aka z/OS
    - UNIX (AIX, Linux, Sun OS, Sun Solaris, HP-UX)
    - Windows
    - iSeries (i5OS)
  - Programming Languages
    - C, COBOL, Java (JNI, WMQ for Java, WMQ for JMS), RPG


Thank
You