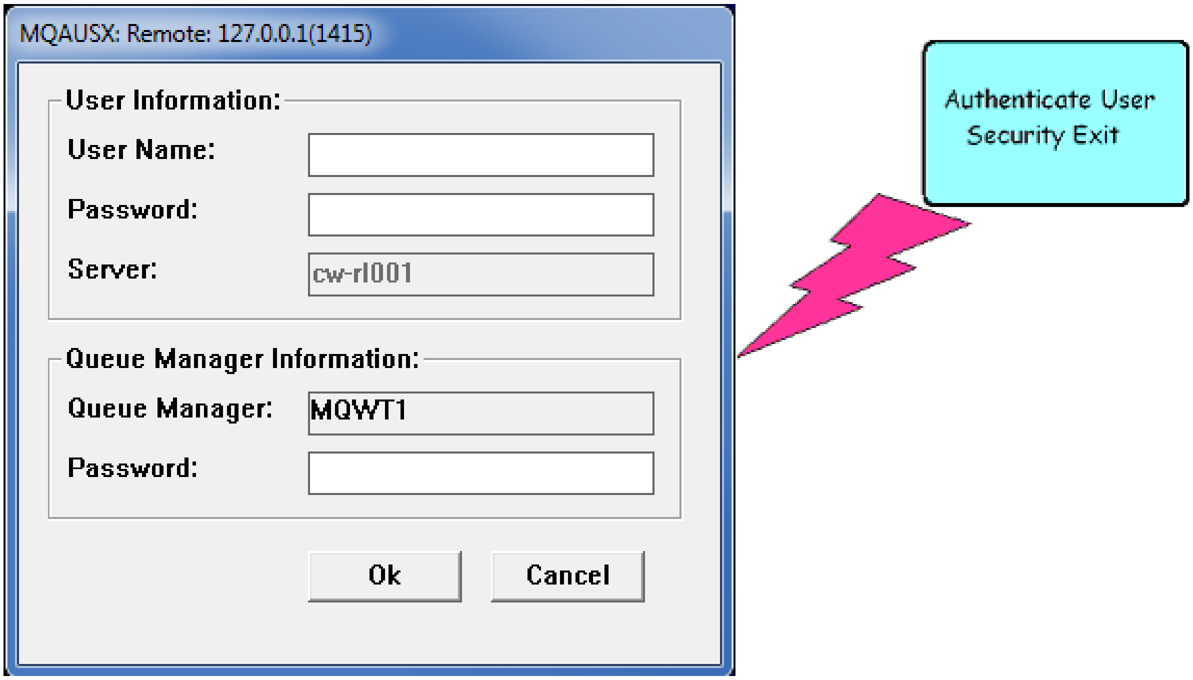
MQ Authenticate User Security Exit Overview



MQAUSX: Remote: 127.0.0.1(1415)

User Information:

User Name:

Password:

Server: cw-rl001

Queue Manager Information:

Queue Manager: MQWT1

Password:

Capitalware Inc.
Unit 11, 1673 Richmond Street, PMB524
London, Ontario N6G2N3
Canada
sales@capitalware.com
https://www.capitalware.com

Last Updated: January 2021.

Table of Contents

1 Introduction

1.1 Overview

MQ Authenticate User Security Exit (MQAUSX) is solution that allows a company to fully authenticate a user who is accessing a IBM MQ resource. It authenticates the user's UserId and Password (and possibly Domain Name) against the server's native OS system, LDAP server, Microsoft's Active Directory, Quest Authentication Services, Centrify's DirectControl, Unix/Linux PAM (Pluggable Authentication Module) or an encrypted MQAUSX FBA file.

The security exit will operate with IBM MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2 in Windows, Unix and Linux environments. It works with Server Connection, Client Connection, Sender, Receiver, Server and Requestor channels of IBM MQ queue manager.

The MQ Authenticate User Security Exit solution is comprised of 2 components: client-side security exit and server-side security exit.

1.1.1 Client-Side Security Exit

The ***client-side security exit*** first checks if the server-side exit is defined for the particular channel. The client-side exit will receive a security token to be used in the encryption process of the user's password. It will prompt the user for his / her UserId and Password (and domain name for Windows), encrypt the data and send it to the server-side security exit.

For each connection attempt, the server-side security exit will verify that it is an acceptable client exit attempting the connection. If so, then the server-side will send a unique security token. When the server-side security exit receives the encrypted data, it will decrypt the incoming data and then perform UserId and Password (and domain) authentication against the native OS system, LDAP server, Microsoft's Active Directory, Quest Authentication Services, Centrify's DirectControl, Unix/Linux PAM (Pluggable Authentication Module) or an encrypted MQAUSX FBA file. If successful, the connection will be allowed.

1.1.2 Server-Side Security Exit

The ***server-side security exit*** supports the concept of 'Proxy IDs'. After a user has been successfully authenticated against the native OS system, LDAP server, Microsoft's Active Directory, Quest Authentication Services, Centrify's DirectControl, Unix/Linux PAM (Pluggable Authentication Module) or an encrypted MQAUSX FBA file and the 'Proxy Mode' flag is set, then the server-side security exit will look up the user's UserID in the Proxy file for their Proxy ID. The Proxy ID will be used for all MQ interactions.

An MQAdmin can define a password for a queue manager. Hence, when enabled, a back-end application and/or end-user would need to not only know their UserID and Password but also the queue manager’s Password to successfully log in. Defining and requiring a queue manager password in MQAUSX is equivalent to adding perimeter security to your system.

The server-side security exit has the ability to allow or restrict users from logging in with the 'mqm' or 'MUSR_MQADMIN' or 'QMQM' UserIDs. This is controlled by the server-side security exit's property keyword 'Allowmqm'.

The server-side security exit has the capability to allow or limit the incoming channel connections according to the name of the associated Server Connection channel (SVRCONN). Each Server Connection channel can be allocated a maximum number of connections and the server-side security exit will ensure that this maximum is not exceeded.

Client connections to a queue manager are limited by either channel name or the 'DefaultMCC' property keyword in the initialization file. In today's use of J2EE applications, it is a possibility that one J2EE application could overwhelm the queue manager with client connections, thus preventing any connections being made from other applications.

The MQAdmin can enable Excessive Client Connections alerting system that counts the number of connections over a period of time (i.e. Day / Hour / Minute) and writes a message to the log when the count exceeds a particular value. If the keyword WriteToEventQueue is set to ‘Y’ then an event message is also written to an event queue. ECC feature is designed to catch applications that are poorly written, for example, applications that continuously connect and disconnect from the queue manager for every message sent or received.

The server-side security exit has the ability to allow or restrict the incoming IP address, hostname and/or SSL DN. The server-side security exit uses a regular expression parser to parse the incoming client IP address, hostname, and/or SSL DN against a predefined regular expression pattern.

The server-side security exit has the ability to allow or restrict the incoming UserID against a group. A list of groups can be queried for the incoming UserID. The groups can be in the local OS or a group file. If MQAUSX is authenticating against an LDAP server then the group querying can be against the LDAP server.

For those channels where authentication is not required, the server-side security exit can be set to not perform this function. This is controlled by the server-side security exit's property keyword 'NoAuth'.

The server-side security exit, when in non-authentication mode, has the ability to allow or restrict users from connecting with a blank UserID value. This is controlled by the server-side security exit's property keyword 'AllowBlankUserID'.

The server-side security exit, when in non-authentication mode, has the ability to allow or restrict the incoming UserID. The server-side security exit uses a regular expression parser to parse the incoming client UserID against a predefined regular expression pattern.

On AIX, HP-UX, Linux, Solaris and Windows, MQAUSX can be configured and used with a non-default installation of MQ in a multi-install MQ environment.

Note: Raspberry Pi is a Linux ARM 32-bit OS (Operating System). Hence, simply follow the Linux 32-bit instructions for installing and using the solution on a Raspberry Pi.

MQAUSX is 4 products in 1

1. If the client application is configured with the client-side security exit then the user credentials are encrypted and sent to the remote queue manager. This is the best level of security.

2. If the client application is not configured with the client-side security exit and the client-side **AND** server-side are at MQ V8 then MQ V8 will encrypt the user credentials as they flow from the client application to the queue manager.

3. If the client application is not configured with the client-side security exit then the user credentials are sent in plain text to the remote queue manager. This feature is available for Java/JMS, Java and C# DotNet client applications. For native applications (i.e. C/C++), then the application must use and populate the MQCSP structure with the UserID and Password.
    - Using MQAUSX with No Client-side Security Exit - Part 1 (coding examples) http://www.capitalware.com/rl_blog/?p=638
    - Using MQAUSX with No Client-side Security Exit - Part 2 (configuring tools like MQ Explorer, SupportPac MO71, MQ Visual Edit, etc..) http://www.capitalware.com/rl_blog/?p=659

4. If the MQAdmin sets the MQAUSX IniFile parameter NoAuth to Y then it functions just like MQ Standard Security Exit (MQSSX). MQSSX does not authenticate. It filters the incoming connection based on UserID, IP address, hostname and/or SSL DN.

1.2 Executive Summary

The ***MQ Authenticate User Security Exit*** solution is comprised of 2 components: client-side security exit and server-side security exit.

1.2.1 Server-Side Security Exit

The server-side security exit is available in 3 forms:

- Windows DLL
- Shared library for AIX, HP-UX, Linux, and Solaris.
- IBM i exit module

The major features of the server-side security exit are as follows:

- Authenticate a user against the server's native OS system, LDAP server, Microsoft's Active Directory, Quest Authentication Services*, Centrify's DirectControl*, PAM* or FBA file.
- Allows or restricts the incoming UserID against a Group
- Provides support for Proxy UserIDs
- Ability to assign a Password to a queue manager for client authentication
- Allows or restricts the incoming IP address against a regular expression pattern
- Allows or restricts the incoming SSL DN against a regular expression pattern
- Allows or restricts the incoming UserID against a regular expression pattern
- Allows or restricts the incoming AD server name against a regular expression pattern**
- Allows or restricts the use of 'mqm', 'MUSER_MQADMIN' or 'QMQM' UserIDs
- Ability to turn off server-side authentication
- Ability to set the maximum number of allowable connections per a given channel (MCC)
- Ability to monitor for excessive client connections (ECC) and then generate an alert
- Provides monitoring tool tie-in by using custom MQ event messages
- Provides logging capability for all connecting client applications regardless if they are successful or not.

* Unix/Linux only
** Windows Only

1.2.2 Client-Side Security Exit

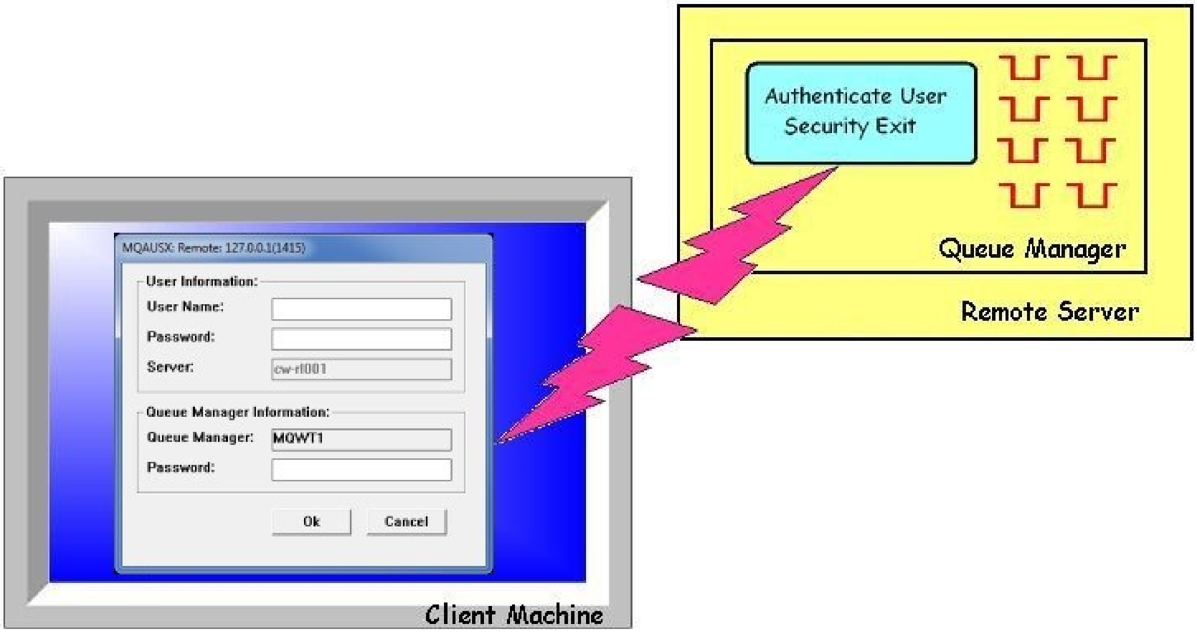
The client-side security exit is available in 5 forms:

- Windows DLL (32-bit & 64-bit)
- Windows DLL for managed .NET (32-bit & 64-bit)
- Java JAR
- Non-GUI shared library for AIX, HP-UX, Linux, and Solaris.
- IBM i exit module

The client-side security exit has been tested against the following MQ client programs:

- IBM's MQ Explorer v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2
- SupportPac MO71 (MQMon)
- IBM's WBIMB Eclipse Tool Kit
- WebSphere Message Broker Explorer V8.0 or higher
- IBM DataPower
- BMC Middleware Management - Administration (BMM Admin)
- BMC's Administration for IBM MQ (AppWatch)
- webMethods MQ Adapter
- Mercury's SiteScope
- Capitalware's MQ Visual Edit
- Capitalware's MQ Visual Browse
- Capitalware's MQ Batch Toolkit
- Capitalware's Universal File Mover
- J2EE application servers i.e. WAS, WebLogic, Jboss, etc...
- Any program that uses Client Channel Tables (i.e. SupportPac MS03, WatchQ, etc.)

1.3 Context Diagram (Logical View)


MQAUSX: Remote: 127.0.0.1(1415)

User Information:

User Name:

Password:

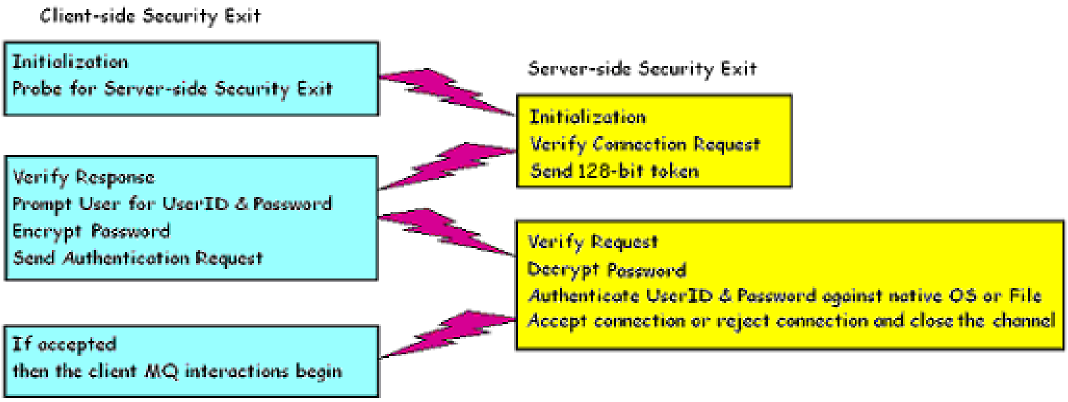
Server: cw-rl001

Queue Manager Information:

Queue Manager: MQWT1

Password:

1.4 Security Message Flow (Logical View)


Client-side Security Exit
Initialization
Probe for Server-side Security Exit
Server-side Security Exit
Initialization
Verify Connection Request
Send 128-bit token
Verify Response
Prompt User for UserID & Password
Encrypt Password
Send Authentication Request
Verify Request
Decrypt Password
Authenticate UserID & Password against native OS or File
Accept connection or reject connection and close the channel
If accepted
then the client MQ interactions begin

1.5 Prerequisites

This section provides the minimum supported software levels. These prerequisites apply to both client-side and server-side installations of MQ Authenticate User Security Exit.

1.5.1 Operating System

MQ Authenticate User Security Exit can be installed on any of the following supported servers:

1.5.1.1 IBM AIX

- IBM AIX 6L version 6.1 or higher

1.5.1.2 HP-UX IA64

- HP-UX v11.23 or higher

1.5.1.3 IBM i (OS/400)

- IBM i V6R1 or higher

1.5.1.4 Linux x86

- Red Hat Enterprise Linux v5, v6, v7, v8
- SUSE Linux Enterprise Server v11, v12, v15

1.5.1.5 Linux x86_64 (64-bit)

- Red Hat Enterprise Linux v5, v6, v7, v8
- SUSE Linux Enterprise Server v11, v12, v15

1.5.1.6 Linux on POWER

- Red Hat Enterprise Linux v5, v6, v7, v8
- SUSE Linux Enterprise Server v11, v12, v15

1.5.1.7 Linux on zSeries (64-bit)

- Red Hat Enterprise Linux v5, v6, v7, v8
- SUSE Linux Enterprise Server v11, v12, v15

1.5.1.8 Raspberry Pi (Linux ARM 32-bit)

- Raspberry Pi OS v9 or higher

1.5.1.9 Sun Solaris

- Solaris SPARC v10 & v11
- Solaris x86_64 v10 & v11

1.5.1.10 Windows

- Windows 2008, 2012 or 2016 Server (32-bit & 64-bit)
- Windows 7, 8, 8.1 & 10 (32-bit & 64-bit)

1.5.2 IBM MQ

- IBM MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2 (both 32-bit and 64-bit)

Operating System	MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2
AIX v6.1 or higher	64-bit
HP-UX IA64 v11.23 or higher	64-bit
IBM i (OS/400)	64-bit
Linux x86	32-bit
Linux x86_64	64-bit
Linux on POWER	64-bit
Linux on zSeries	32-bit & 64-bit
Raspberry Pi ARM	32-bit
Solaris SPARC v8 or higher	64-bit
Solaris x86_64 v10	64-bit
Windows 2008, 2012, 2016, 7, 8, 8.1 & 10	32-bit & 64-bit

1.5.3 LDAP Server

- Microsoft's Active Directory for Windows 2000 Server or higher
- Novell's eDirectory v8 or higher
- Oracle 9i Internet Directory or higher
- OpenLDAP v2.1 or higher
- IBM Tivoli Directory Server
- z/OS Integrated Security Services LDAP Server v1.6 or higher

1.5.3.1 Dependencies

- An LDAP client needs to be installed on the same server as the MQ queue manager.
- To use SSL, the LDAP server and the LDAP client must be configured for SSL.

1.5.4 Windows 64-bit

The following is the software prerequisite for Windows 64-bit:

- Microsoft Visual C++ 2010 Redistributable Package (x64)
  https://www.microsoft.com/en-ca/download/details.aspx?id=14632

2 Client-side Security Exit Tested Applications

This section describes on what applications has been tested with the client-side security exit.

The client-side security exit has been built and tested on the following operating systems:

1. Java v1.5 or higher on platforms that support a JVM of v1.5 or higher
2. Windows 2003 / 2008 / 2012 / 7 / 8.0 / 8.1 / 10 native GUI and non-GUI mode
3. J2EE applications native non-GUI mode
4. .NET environment both managed and unmanaged
5. AIX v5.1 or higher native non-GUI mode
6. HP-UX v11 or higher native non-GUI mode
7. Solaris v8 or higher native non-GUI mode
8. RHEL v4 or higher & SLES v10 or higher native non-GUI mode

The client-side security exit has been tested with the following applications:

1. IBM's MQ Explorer v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2
2. SupportPac MO71 (MQMon)
3. IBM's WBIMB Eclipse Tool Kit
4. WebSphere Message Broker Explorer V8.0 or higher
5. IBM DataPower
6. BMC Middleware Management - Administration (BMM Admin)
7. BMC's Administration for IBM MQ (AppWatch)
8. webMethods MQ Adapter
9. WebSphere Application Server
10. WebLogic
11. JBoss
12. Mercury's SiteScope
13. Universal File Mover
14. Capitalware's MQ Visual Edit
15. Capitalware's MQ Visual Browse
16. Capitalware's MQ Batch Toolkit
17. Capitalware's Universal File Mover
18. Any program that uses Client Channel Tables (i.e. SupportPac MS03, WatchQ, etc.)

3 Appendix A – Encryption

MQ Authenticate User Security Exit Solution uses the Advanced Encryption Standard (AES) for encryption and decryption of the user's password between the client-side security exit and the server-side security exit.

Wikipedia

> *the Advanced Encryption Standard (AES) is an encryption standard adopted by the U.S. government. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as Rijndael. Each AES cipher has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively. The AES ciphers have been analyzed extensively and are now used worldwide, as was the case with its predecessor,[3] the Data Encryption Standard (DES).*
>
> *AES was announced by National Institute of Standards and Technology (NIST) as U.S. FIPS PUB 197 (FIPS 197) on November 26, 2001 after a 5-year standardization process in which fifteen competing designs were presented and evaluated before Rijndael was selected as the most suitable (see Advanced Encryption Standard process for more details). It became effective as a Federal government standard on May 26, 2002 after approval by the Secretary of Commerce. It is available in many different encryption packages. AES is the first publicly accessible and open cipher approved by the NSA for top secret information*