QueueMetrix Gatekeeper

Overview

Gatekeeper is a security plugin for MQ and provides a secure way for clients to connect to an MQ queue manager. It uses a client authentication exit module to extend the functionality of MQ to provide a method for JMS and other types of client connections to be authenticated using standard LDAP Simple authentication.

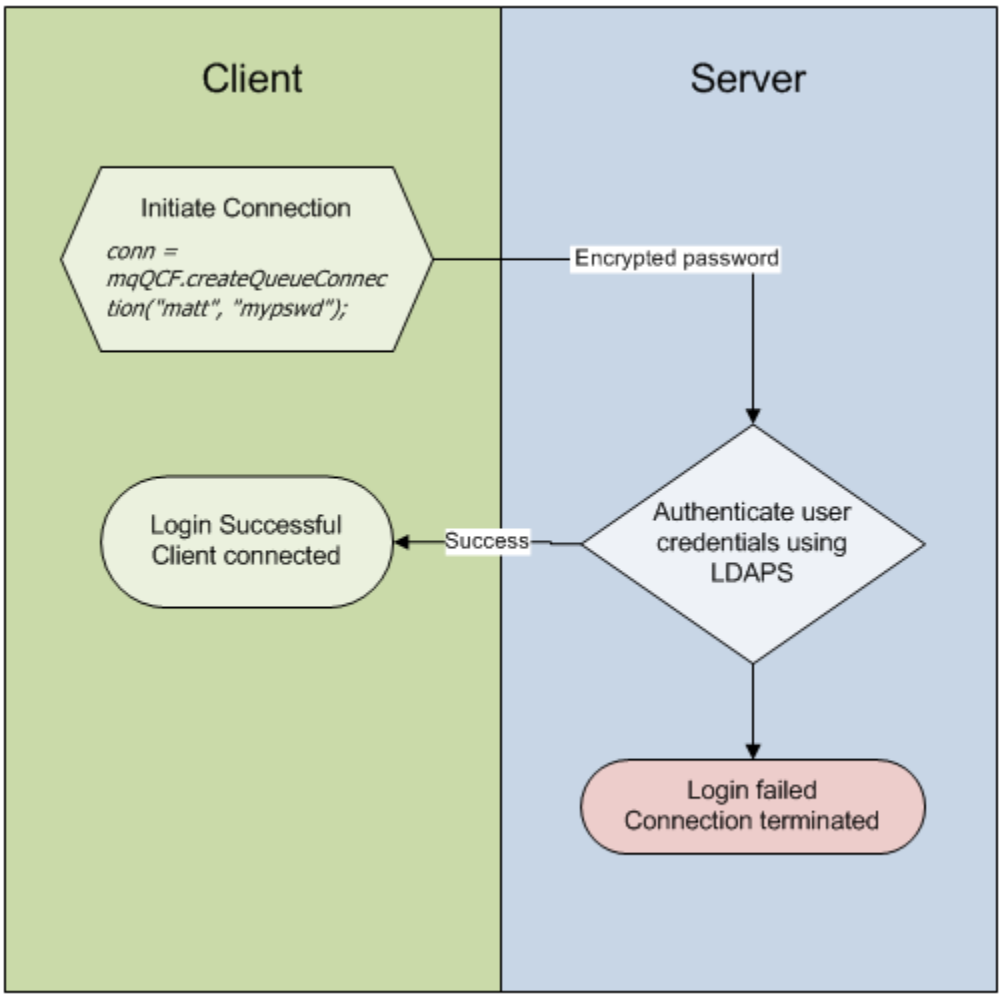
The module itself is called a 'security channel exit' and is named *libMQGatekeeper*. The module is deployed to an MQ server and is used to protect client MQ connections by providing username and password authentication against an enterprises single sign-on (SSO) such as LDAPS or Microsoft's Active Directory.

Client passwords are protected during channel authentication by using standard MQ one way SSL encryption.

The module provides a number of key security features such as,

- Username/password authentication performed using LDAP/S simple authentication.
- Every channel can employ a different security profile
- Auto fail-over to alternate LDAP/S server when one is not available
- Supports Microsoft Active Directory (AD) LDAP
- One, or two way SSL on the connecting MQ client channel to protect the password on the wire.
- Supports LDAP group memberships such as an AD group
- IP address filtering *(backward compatible with BlockIP2 rules file)*
- Client user id translation/pass-through for object level authorisation (OAM)
- Multiple client API support

Diagram showing how the connection is established



Source Code

Located in GitHub: https://github.com/queuemetrix/mqgatekeeper.git

- Building MQ Client LDAP Authentication Exit
- MQ LDAP Properties file - ldap.properties

Examples

WebSphere MQ base Java

```
MQEnvironment.hostname = "10.10.10.10(1414)";
MQEnvironment.channel = "LAMAXU_CHANNEL";
MQEnvironment.userID = "lamaxu";
MQEnvironment.password = "mypswd";
MQQueueManager _qMgr = new
MQQueueManager("DEMO");
```

WebSphere MQ base JMS

```
mqQCF = new MQQueueConnectionFactory();
mqQCF.setQueueManager("DEMO");
mqQCF.setHostName("10.10.10.10(1414)");
mqQCF.setChannel("LAMAXU.CHANNEL");
mqQCF.setTransportType(JMSC.MQJMS_TP_CLIENT_MQ_TCPIP);
conn = mqQCF.createQueueConnection("lamaxu",
"mypswd");
```

C# .Net

```
MQEnvironment.Hostname = "10.10.10.10(1414)";
MQEnvironment.Channel = "LAMAXU.CHANNEL";
MQEnvironment.UserId = "lamaxu";
MQEnvironment.Password = "mypswd";
MQQueueManager _qMgr = new
MQQueueManager("DEMO");
```

C

```
char QMName[MQ_Q_MGR_NAME_LENGTH+1|MQ_Q_MGR_NAME_LENGTH+1] = "TESTQM";
char UserId[64] = "lamaxu";
char Password[64] = "mypswd";
strncpy(ClientConn.ConnectionName, "192.168.1.1(1414)",
MQ_CONN_NAME_LENGTH);
strncpy(ClientConn.ChannelName, "LAMAXU.CHANNEL",
MQ_CHANNEL_NAME_LENGTH);
mqCSP.AuthenticationType = MQCSP_AUTH_USER_ID_AND_PWD;
mqCSP.CSPUserIdPtr = &UserId;
mqCSP.CSPUserIdLength = strlen(UserId);
mqCSP.CSPPasswordPtr = &Password;
mqCSP.CSPPasswordLength = strlen(Password);
ConnectOptions.SecurityParmsPtr = &mqCSP;
ConnectOptions.ClientConnPtr = &ClientConn;
ConnectOptions.Version = MQCNO_VERSION_2;
MQCONNX (QMName, &ConnectOptions, &HConn, &CompCode, &Reason);
```

C++

```
pchannel = new ImqChannel;
pchannel -> setConnectionName("10.10.10.10(1414)");
pchannel -> setChannelName("LDAPAUTH_CHANNEL");
pchannel -> setTransportType( MQXPT_TCP );
pchannel -> setUserId( "matt" );
pchannel -> setPassword( "mypswd" );
mgr.setName(QMName);
mgr.setChannelReference( pchannel );
if ( mgr.connect( ) ) { }
```

Related articles

- QueueMetrix Gatekeeper

Building MQ Gatekeeper

Build Instructions

First check out the code from GitHub, https://github.com/queuemetrix/mqgatekeeper.git

> **Note,** you may need to adjust the Makefiles to enter the correct paths for the dependency include and lib directories.

Linux Example;

> *bash-4.1$ pwd*
> */opt/mqm/src/c/libMQAuthLdap*
>
> *bash-4.1$ make clean*
> */bin/rm -f libMQAuthLdap*
> */bin/rm -f* ***.o *.tar *.tar.Z *.ba core x*** **.uu *.trc *.log*
>
> *bash-4.1$ make*
> *gcc -DLINUX -m64 -shared -fPIC -DANSI_PROTOTYPES -D_REENTRANT -I/opt/mqm/inc/*
> *-I/opt/mqm/scripts/linux64/openssl/include/ -I/opt/mqm/scripts/linux64/openldap/include -Wl,-rpath=/opt/mqm/lib64*
> *-Wl,-rpath=/opt/mqm/scripts/linux64/openldap/lib -Wl,-rpath=/opt/mqm/scripts/linux64/openssl/lib -Wl,-rpath=/usr/lib64 -c*
> *libMQAuthLdap.c*
> *gcc -DLINUX -m64 -shared -fPIC -DANSI_PROTOTYPES -D_REENTRANT -I/opt/mqm/inc/*
> *-I/opt/mqm/scripts/linux64/openssl/include/ -I/opt/mqm/scripts/linux64/openldap/include -Wl,-rpath=/opt/mqm/lib64*
> *-Wl,-rpath=/opt/mqm/scripts/linux64/openldap/lib -Wl,-rpath=/opt/mqm/scripts/linux64/openssl/lib -Wl,-rpath=/usr/lib64 -c*
> *LdapAuthenticateUser.c*
> *gcc -DLINUX -m64 -shared -fPIC -DANSI_PROTOTYPES -D_REENTRANT -I/opt/mqm/inc/*
> *-I/opt/mqm/scripts/linux64/openssl/include/ -I/opt/mqm/scripts/linux64/openldap/include -Wl,-rpath=/opt/mqm/lib64*
> *-Wl,-rpath=/opt/mqm/scripts/linux64/openldap/lib -Wl,-rpath=/opt/mqm/scripts/linux64/openssl/lib -Wl,-rpath=/usr/lib64 -c*
> *GetProperties.c*
> *gcc -DLINUX -m64 -shared -fPIC -DANSI_PROTOTYPES -D_REENTRANT -I/opt/mqm/inc/*
> *-I/opt/mqm/scripts/linux64/openssl/include/ -I/opt/mqm/scripts/linux64/openldap/include -Wl,-rpath=/opt/mqm/lib64*
> *-Wl,-rpath=/opt/mqm/scripts/linux64/openldap/lib -Wl,-rpath=/opt/mqm/scripts/linux64/openssl/lib -Wl,-rpath=/usr/lib64 -c*
> *CheckIPAddress.c*
> *gcc -DLINUX -m64 -shared -fPIC -DANSI_PROTOTYPES -D_REENTRANT -I/opt/mqm/inc/*
> *-I/opt/mqm/scripts/linux64/openssl/include/ -I/opt/mqm/scripts/linux64/openldap/include -Wl,-rpath=/opt/mqm/lib64*
> *-Wl,-rpath=/opt/mqm/scripts/linux64/openldap/lib -Wl,-rpath=/opt/mqm/scripts/linux64/openssl/lib -Wl,-rpath=/usr/lib64*
> *-L/opt/mqm/lib64 -L/opt/mqm/scripts/linux64/openssl/lib -L/opt/mqm/scripts/linux64/openldap/lib -lmqm_r -lmqmcs_r -lssl -lcrypt*
> *-lldap_r libMQAuthLdap.o -o libMQAuthLdap *
> *LdapAuthenticateUser.o *
> *GetProperties.o *
> *CheckIPAddress.o*
>
> *bash-4.1$ make install*
> *cp libMQAuthLdap /var/mqm/exits64*

Building the Dependencies

The LDAP authentication exit *libMQAuthLdap* has a number of dependencies that need to be built, or available, when the exit is compiled.

These are;

- OpenLdap
- OpenSSL

Building OpenLdap

Install OpenSSL

> ***Linux Example***
> *mkdir /opt/mqm/scripts/linux64/openssl*
> *cd /opt/mqm/scripts/linux64/openssl-1.0.1c*
> *./config --prefix=/opt/mqm/scripts/linux64/openssl enable-tlsext shared*
> *make*
> *make install*

> ***Solaris SPARC-64 Example***
> *mkdir /opt/mqm/scripts/sparc/openssl*
> *cd /opt/mqm/scripts/linux64/openssl-1.0.1c*
> *./Configure solaris64-sparcv9-cc --prefix=/opt/mqm/scripts/sparc/openssl enable-tlsext shared*
> *make*
> *make install*
>
> ***AIX 64-bit version with IBM XL C***
> *./Configure threads --prefix=/usr/local/security/openssl aix64-cc shared*

```
make
make install
```

Install OpenLdap

RHEL 6 Example

```
mkdir /opt/mqm/scripts/linux64/openldap
cd /opt/mqm/scripts/linux64/openldap-2.4.32
export LD_LIBRARY_PATH=/opt/mqm/scripts/linux64/berkeley_db/lib:$LD_LIBRARY_PATH
env CPPFLAGS="-I/opt/mqm/scripts/linux64/openssl/include LDFLAGS="-L/opt/mqm/scripts/linux64/openssl/lib
-L/opt/mqm/scripts/linux64/berkeley_db/lib" \
./configure --with-tls=openssl --enable-shell --prefix=/opt/openldap --disable-bdb --disable-hdb --disable-static --enable-dynamic
--disable-slapd --disable-debug
make depend
make install
```

Solaris x86-64 Example

```
mkdir /opt/mqm/scripts/solx86/openldap-2.4.32
cd /opt/mqm/scripts/solx86/src-openldap-2.4.32
export LDFLAGS="-m64 -L/opt/mqm/scripts/solx86/openssl/lib"
export CFLAGS="-m64 -I/opt/mqm/scripts/solx86/openssl/include"
export CPPFLAGS="-m64 -I/opt/mqm/scripts/solx86/openssl/include"

./configure --with-tls=openssl --enable-shell --prefix=/opt/openldap --disable-bdb --disable-hdb --disable-static --enable-dynamic
--disable-slapd --disable-debug
make depend
make install
```

AIX Example

```
mkdir /opt/openldap-2.4.42
cd /opt/src-openldap-2.4.42
#export LD_LIBRARY_PATH=/opt/mqm/scripts/solx86/berkeley_db/lib:$LD_LIBRARY_PATH
export LDFLAGS="-q64 -L/opt/openssl/lib"
export CPPFLAGS="-q64 -I/opt/openssl/include"

 export OBJECT_MODE=64
./configure --with-tls=openssl --enable-shell --prefix=/opt/openldap --disable-bdb --disable-hdb AR="ar -X64" --disable-static
--enable-dynamic --disable-slapd --disable-debug
make depend
make install
```

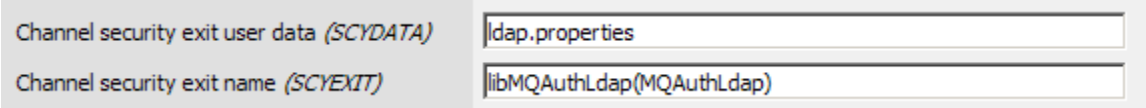
QueueMetrix Gatekeeper Properties file

Properties file

Defining the Exit

The ***ldap.properties*** file is used by the MQ ***libMQAuthLdap*** security exit library to specify the security options for a particular channel. A common ***ldap.properties*** file can be used for all channels or a new one created per channel. The property file is specified as part of the MQ channel definition.

Example;



Example ldap.properties file

- QMGR
- CHANNEL
- LogFilePath
- LogFileTag
- PROP_AUTHENTICATE_USER
- Debug
- PROP_CHECK_HOST_ADDRESS

- PROP_HOST_ADDRESS_RULE_FILE_FORMAT
- PROP_HOST_ADDRESS_RULE_FILE_NAME
- PROP_LDAP_AUTH_METHOD
- PROP_LDAP_VERSION
- PROP_LDAP_SCOPE
- PROP_LDAP_SCOPE_SUBTREE
- PROP_LDAP_SERVER_URL
- PROP_LDAP_PRINCIPAL_PREFIX
- PROP_LDAP_PRINCIPAL_SUFFIX
- PROP_LDAP_BASE_DN
- PROP_LDAP_GROUP_SEARCH_FILTER
- PROP_LDAP_USER_SEARCH_ATTRIBUTE
- PROP_LDAP_OPT_X_TLS_CACERTFILE
- PROP_LDAP_OPT_X_TLS_CACERTDIR
- PROP_LDAP_OPT_X_TLS_REQUIRE_CERT
- PROP_LDAP_OPT_REFERRALS
- PROP_LDAP_OPT_TIMELIMIT
- PROP_LDAP_OPT_NETWORK_TIMEOUT

Example Properties file

```
# Authenticate against the AD Domain controllers
QMGR=QM1
CHANNEL=LDAPAUTH_CHANNEL
LogFilePath=/var/mqm/errors/
LogFileTag=AD

PROP_AUTHENTICATE_USER=TRUE
PROP_CHECK_HOST_ADDRESS=TRUE
PROP_HOST_ADDRESS_RULE_FILE_FORMAT=BIP2
PROP_HOST_ADDRESS_RULE_FILE_NAME=/var/mqm/exits/QM1

PROP_LDAP_AUTH_METHOD=LDAP_AUTH_SIMPLE
PROP_LDAP_VERSION=LDAP_VERSION3
PROP_LDAP_SCOPE=LDAP_SCOPE_SUBTREE
PROP_LDAP_SERVER_URL=ldaps://server123 ldaps://server456
PROP_LDAP_PRINCIPAL_PREFIX=NTDOMAIN\
PROP_LDAP_PRINCIPAL_SUFFIX=
PROP_LDAP_BASE_DN=DC=pc,DC=internal,DC=queuemetrix,DC=com
PROP_LDAP_GROUP_SEARCH_FILTER=(|(memberOf=CN=QueueMetrix
Team,OU=Groups,OU=Exchange,DC=pc,DC=internal,DC=queuemetrix,DC=com))
PROP_LDAP_USER_SEARCH_ATTRIBUTE=sAMAccountName
PROP_LDAP_OPT_X_TLS_CACERTFILE=
PROP_LDAP_OPT_X_TLS_CACERTDIR=/etc/ssl/certs/
PROP_LDAP_OPT_X_TLS_REQUIRE_CERT=LDAP_OPT_X_TLS_NEVER
PROP_LDAP_OPT_REFERRALS=LDAP_OPT_OFF
PROP_LDAP_OPT_TIMELIMIT=3
PROP_LDAP_OPT_NETWORK_TIMEOUT=30

# Authenticate against the LDS server
QMGR=QM1
CHANNEL=LDAPAUTH_CHANNEL
LogFilePath=/var/mqm/errors/
LogFileTag=LDS

PROP_AUTHENTICATE_USER=TRUE
PROP_CHECK_HOST_ADDRESS=TRUE
PROP_HOST_ADDRESS_RULE_FILE_FORMAT=BIP2
```

```
PROP_HOST_ADDRESS_RULE_FILE_NAME=/var/mqm/exits/QM1
PROP_LDAP_AUTH_METHOD=LDAP_AUTH_SIMPLE
PROP_LDAP_VERSION=LDAP_VERSION3
PROP_LDAP_SCOPE=LDAP_SCOPE_SUBTREE
PROP_LDAP_SERVER_URL=ldaps://server123
PROP_LDAP_PRINCIPAL_PREFIX=CN=
PROP_LDAP_PRINCIPAL_SUFFIX=,OU=userProxy,DC=LDS,DC=Internal,DC=queuemetrix
,DC=com
PROP_LDAP_BASE_DN=OU=Users,DC=LDS,DC=Internal,DC=queuemetrix,DC=com
PROP_LDAP_GROUP_SEARCH_FILTER=(|(memberOf=CN=QueueMetrix
Team,OU=ADGroups,DC=LDS,DC=Internal,DC=queuemetrix,DC=com))
PROP_LDAP_USER_SEARCH_ATTRIBUTE=sAMAccountName
PROP_LDAP_OPT_X_TLS_CACERTFILE=
PROP_LDAP_OPT_X_TLS_CACERTDIR=/etc/ssl/certs/
PROP_LDAP_OPT_X_TLS_REQUIRE_CERT=LDAP_OPT_X_TLS_TRY
PROP_LDAP_OPT_REFERRALS=LDAP_OPT_OFF
PROP_LDAP_OPT_TIMELIMIT=3
PROP_LDAP_OPT_NETWORK_TIMEOUT=30

# Another Channel without IP filtering
QMGR=QM1
CHANNEL=LDAPAUTH_CHANNEL
LogFilePath=/var/mqm/errors/
LogFileTag=Multigroup

PROP_AUTHENTICATE_USER=TRUE
PROP_CHECK_HOST_ADDRESS=FALSE
PROP_HOST_ADDRESS_RULE_FILE_FORMAT=BIP2
PROP_HOST_ADDRESS_RULE_FILE_NAME=/var/mqm/exits/QM1

PROP_LDAP_AUTH_METHOD=LDAP_AUTH_SIMPLE
PROP_LDAP_VERSION=LDAP_VERSION3
PROP_LDAP_SCOPE=LDAP_SCOPE_SUBTREE
PROP_LDAP_SERVER_URL=ldaps://server123
PROP_LDAP_PRINCIPAL_PREFIX=CN=
PROP_LDAP_PRINCIPAL_SUFFIX=,OU=userProxy,DC=LDS,DC=Internal,DC=queuemetrix
,DC=com
PROP_LDAP_BASE_DN=OU=Users,DC=LDS,DC=Internal,DC=queuemetrix,DC=com
PROP_LDAP_GROUP_SEARCH_FILTER=(|(memberOf=CN=QueueMetrix
Team,OU=ADGroups,DC=LDS,DC=Internal,DC=queuemetrix,DC=com)(memberOf=CN=Que
ueMetrix Admin,OU=ADGroups,DC=LDS,DC=Internal,DC=queuemetrix,DC=com))
PROP_LDAP_USER_SEARCH_ATTRIBUTE=sAMAccountName
PROP_LDAP_OPT_X_TLS_CACERTFILE=
PROP_LDAP_OPT_X_TLS_CACERTDIR=/etc/ssl/certs/
PROP_LDAP_OPT_X_TLS_REQUIRE_CERT=LDAP_OPT_X_TLS_TRY
PROP_LDAP_OPT_REFERRALS=LDAP_OPT_OFF
PROP_LDAP_OPT_TIMELIMIT=3
PROP_LDAP_OPT_NETWORK_TIMEOUT=30

# And another Channel
QMGR=QM1
CHANNEL=OTHER_CHANNEL
LogFilePath=/var/mqm/errors/
```

```
LogFileTag=IP_Only
PROP_AUTHENTICATE_USER=FALSE
PROP_CHECK_HOST_ADDRESS=TRUE
```

```
PROP_HOST_ADDRESS_RULE_FILE_FORMAT=BIP2
PROP_HOST_ADDRESS_RULE_FILE_NAME=/var/mqm/exits/QM1
```

QMGR

This is the name of the queue manager that the subsequent config will be used for.

Example;

> *QMGR=QM1*

CHANNEL

This is the name of the channel that the subsequent config will be used for.

Example;

> *CHANNEL=LDAPAUTH_CHANNEL*

LogFilePath

This is the directory where the error logs will be written to.

Example;

> *LogFilePath=/var/mqm/error/*

LogFileTag

This is the name tagged to the error log after the channel name.

Example;

> *LogFileTag=mylogfile*

PROP_AUTHENTICATE_USER

If this is set to FALSE then the username & password of connecting client will NOT be authenticated on the LDAP. If any other value is set then the default option is TRUE.

Example;

> *PROP_AUTHENTICATE_USER=FALSE*

Debug

This value determines whether debug messages will be written to the exit log.

Valid options

> *0 = Minimal messages*
> *1 = Full debug logging*

PROP_CHECK_HOST_ADDRESS

If this is set to TRUE then the IP address of the connecting client will be checked. If any other value is set then the option is disabled.

Example;

> *PROP_CHECK_HOST_ADDRESS=TRUE*

PROP_HOST_ADDRESS_RULE_FILE_FORMAT

This specifies the format of the IP address rules file. At present only the BIP2 *(BlockIP2)* file is supported to provide compatibility with the existing IP filter rule files.

Example;

> *PROP_HOST_ADDRESS_RULE_FILE_FORMAT=BIP2*

PROP_HOST_ADDRESS_RULE_FILE_NAME

This option sets the location and name of the IP address rules file.

Example;

> *PROP_HOST_ADDRESS_RULE_FILE_NAME=/var/mqm/exits/QM1*

PROP_LDAP_AUTH_METHOD

This value is always set as LDAP_AUTH_SIMPLE

Valid Options

> *LDAP_AUTH_SIMPLE*

PROP_LDAP_VERSION

This sets the LDAP version to use and should generally be set to LDAP_VERSION3

Valid Options

> *LDAP_VERSION1*
> *LDAP_VERSION2*
> *LDAP_VERSION3*

PROP_LDAP_SCOPE

> *Not used yet'*

PROP_LDAP_SCOPE_SUBTREE

> *Not used yet*

PROP_LDAP_SERVER_URL

This is the ldap: or lpdaps: server URL

Note that multiple servers can be specified for redundancy purposes delimited by a space char.

Example;

On the LDS

> *PROP_LDAP_SERVER_URL=ldaps://server123.queuemetrix.com*

and for the AD domain controller

> *PROP_LDAP_SERVER_URL=ldaps://server123 ldaps://server456*

PROP_LDAP_PRINCIPAL_PREFIX

This is the prefix of the connecting user

Example;

On the LDS

> *PROP_LDAP_PRINCIPAL_PREFIX=CN=*

and for the AD domain controller

> *PROP_LDAP_PRINCIPAL_PREFIX=NTDOMAIN*

PROP_LDAP_PRINCIPAL_SUFFIX

This is the base DN of the connecting user

Example;

On the LDS

> *PROP_LDAP_PRINCIPAL_SUFFIX=,OU=userProxy,DC=LDS,DC=Internal,DC=queuemetrix,DC=com*

and for the AD domain controller

> *PROP_LDAP_PRINCIPAL_SUFFIX=*

PROP_LDAP_BASE_DN

This is the base DN used when searching for user group memberships

Example;

On the LDS

> *PROP_LDAP_BASE_DN=OU=Users,DC=LDS,DC=Internal,DC=queuemetrix,DC=com*

and for the AD domain controller

> *PROP_LDAP_BASE_DN=DC=pc,DC=internal,DC=queuemetrix,DC=com*

PROP_LDAP_GROUP_SEARCH_FILTER

This value contains an LDAP filter string and is used to check for membership of one or more AD groups or any other valid LDAP search options.

Example;

> *(|(memberOf=CN=QueueMetrix Team,OU=ADGroups,DC=LDS,DC=Internal,DC=queuemetrix,DC=com)(memberOf=CN=QueueMetrix Team,OU=ADGroups,DC=LDS,DC=Internal,DC=queuemetrix,DC=com))*

PROP_LDAP_USER_SEARCH_ATTRIBUTE

This is the ldap attribute that contains the user name to be filtered on. The sAMAccountName is for AD (Active Directory) authentication.

Example;

> *PROP_LDAP_USER_SEARCH_ATTRIBUTE=sAMAccountName*

PROP_LDAP_OPT_X_TLS_CACERTFILE

This is the path to the CA Certificate .pem file

Example;

> *PROP_LDAP_OPT_X_TLS_CACERTFILE=/etc/ssl/certs/root.pem*

PROP_LDAP_OPT_X_TLS_CACERTDIR

This is the path to the CA Certificate directory store

Example;

> *PROP_LDAP_OPT_X_TLS_CACERTDIR=/etc/ssl/certs/*

PROP_LDAP_OPT_X_TLS_REQUIRE_CERT

This option determines how the certificate from the lpdaps: service is handled

Valid options are;

> *LDAP_OPT_X_TLS_NEVER*
> *LDAP_OPT_X_TLS_HARD*
> *LDAP_OPT_X_TLS_DEMAND*
> *LDAP_OPT_X_TLS_ALLOW*
> *LDAP_OPT_X_TLS_TRY*

PROP_LDAP_OPT_REFERRALS

This option sets whether to follow LDAP referals during searches. It MUST be set to LDAP_OPT_OFF when authenticating against the AD domain controllers.

Valid options are;

> *LDAP_OPT_OFF*
> *LDAP_OPT_ON*

PROP_LDAP_OPT_TIMELIMIT

This option sets the LDAP search timeout in seconds

Example;

> *PROP_LDAP_OPT_TIMELIMIT=3*

PROP_LDAP_OPT_NETWORK_TIMEOUT

This option sets the network timeout in seconds

Example;

> *PROP_LDAP_OPT_NETWORK_TIMEOUT=30*