Tivoli Netcool Support's Guide to Object Server SSL/TLS Configuration

by

Jim Hutchinson

Document release: 2.0

Table of Contents

1 Introduction

1.1 Overview

There are many ways to configure Netcool/OMNIbus 8.1 for secure SSL/TLS communications. The Netcool/OMNIbus 8.1 discusses the best practice method, providing copy and paste boxes to assist customers with the configuration of the servers and gateways.

This document is designed to discuss a specific configuration and how to proceed with checking its configuration as an example for other configurations. Where possible, best practice configuration's are applied for integration and security. Please refer to the product manual for current valid details.

The configurations discussed are for adding TLS/SSL support to a running multi-tier system.

To simplify the configuration of the aggregation layer a single CA and AGG_V certificate is used, with the $NCHOME/etc/security/keys directory being a copy of each other on the aggregation layer.

For simplicity only the aggregation and collection layers are considered.

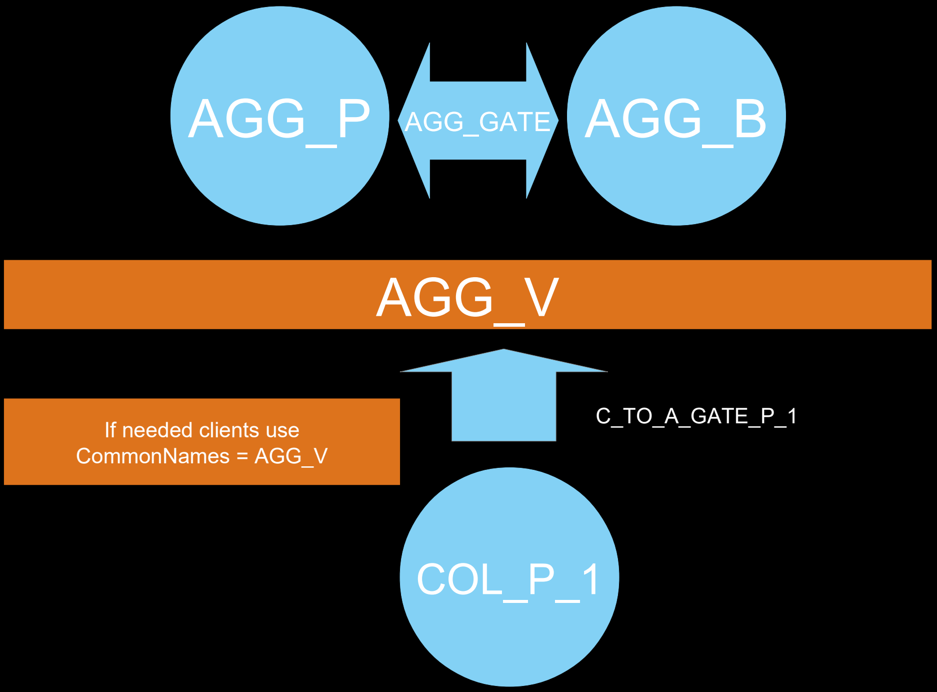
The scripts and examples provided are designed for classroom usage rather than production, with the details required for production usage being defined by your companies security team.

For example, in a production environment the signing of certificates can be performed using a corporate or third part certificate authority.

The use of SSL/TLS is tied with the use of FIPS, and the AES_FIPS setting for probes and gateways. The default Netcool/OMNIbus SSL security settings are not at the FIPS standard. Enabling FIPS in a Netcool/OMNIbus installation enforces a security standard higher than default, therefore certain criteria are required above what is set by default.

1.2 Netcool/OMNIbus system diagram

1



1.3 Netcool/OMNIbus TLS/SSL Settings

The main certificate keystore is $NCHOME/etc/security/keys.

Certificates have a given label in the keystore and a Common Name on the certificates subject.
Typically the Object Server Name property defines the certificates label and Common Name, unless there is a requirement to use a virtual object server name, in which case there is the option to use this name for object servers certificate using the Ipc.SSLCertificateLabel property setting.

For example with AGG_V.

The Netcool/OMNIbus clients allow the object server certificates to be referenced using the certificates Common Name.

Object Server property file settings
Defines the label of the object server certificate when it is not the object server Name
```
Ipc.SSLCertificateLabel : ''
```

Bi-directional gateway property file settings
```
Gate.ObjectServerA.CommonNames: ''
Gate.ObjectServerB.CommonNames: ''
```

Un-directional gateway property file settings
```
Gate.Reader.CommonNames: ''
Gate.Writer.CommonNames: ''
```

Probe Property file settings
```
SSLServerCommonName: ''
```

1.4 Object Server settings

The main object server property setting is the Ipc.SSLCertificateLabel property, which is only required if the certificate is being linked to the Virtual Object Server, otherwise the certificate label used, is the object server name.

Example property settings.

```
# Object Server name
Name                   : 'AGG_P'
# SSL/TLS
Ipc.SSLCertificateLabel: 'AGG_V'
# AES
PasswordEncryption     : 'AES'
# AES_FIPS
ConfigCryptoAlg        : 'AES_FIPS'
ConfigKeyFile          : '$NCHOME/etc/security/crypt/objectserver.keyfile'
# Typical usage of AES_FIPS encryption
PA.Name                : 'NCO_PA'
PA.Username            : 'root'
PA.Password            : '@44:1IffU8iSzJdW3Vq35Gj/wmqpPcI0zaN+flmtykEvRqI=@'
```

1.5 nco_sql usage

Both nco_sql and nco_ping use the details defined in the local Netcool/OMNIbus interfaces file, which is created using nco_igen from the $NCHOME/etc/omni.dat file.

Neither support the use of a label property, or a CommonNames setting, and are unable to connect to object servers whose SSL certificate label is not the object server name being used at the command line.

1.6 Enforcing FIPS and TLS

The fips.conf informs the clients and servers to enforce the FIPS configuration.
Defining additional parameters inside the fips.conf file sets additional requirements.
The sslciphers.conf file can be used to disable unwanted insecure SSL/TLS versions.

File : $NCHOME/etc/security/fips.conf

```
SP800_131MODE=TRUE
TLS12_ONLY=TRUE
```

File : $NCHOME/etc/security/sslciphers.conf

```
sslv3_remove=_
tls10_remove=_
tls11_remove=_
```

Note that if these files are used to define the security level, the files need to be set consistently and the version of Netcool/OMNIbus needs to be identical within the system.

The Java product allows SSL/TLS ciphers to be disabled too.

The latest Netcool/OMNIbus Java 8 disables SSLv1, TLSv1, TLSv1.1 by default.

File : $NCHOME/platform/linux2x86/jre64_1.8.0/jre/lib/security/java.security

```
jdk.tls.disabledAlgorithms=SSLv3, TLSv1, TLSv1.1, RC4, DES, MD5withRSA, DH keySize <
1024, DESede, \ EC keySize < 224, 3DES_EDE_CBC, anon, NULL, DES_CBC
```

1.7 The omni.dat file

The omni.dat file defines how object servers are addressed.

For local object servers and gateways, their listening ports.

```
# Virtual Object Server - VOS
[AGG_V]
{
# For nco_sql usage on primary
Primary:  host1 ssl 8100
# For nco_sql usage on secondary
# Primary: host2 8100
# For AGG_V usage
# Primary: host1 8100
# Backup:  host2 ssl 8100
}
# Primary object server
[AGG_P]
{
Primary: host1 ssl 8100
}
# Secondary object server
[AGG_B]
{
Primary: host2 ssl 8100
}
# Bi-directional gateway
[AGG_GATE]
{
        Primary: localhost 8300
}
# Oracle gateway
[G_JORA]
{
        Primary: localhost 8301
}
# EOF
```

1.8 Bi-directional gateway settings

When the object server name is not being used for the Common Name, the aggregation bi-directional object server can use the property settings to define the object servers certificates   Common Name.

```
Gate.ObjectServerA.Server        : 'AGG_P'
Gate.ObjectServerA.CommonNames   : 'AGG_V'
ObjectServerB.Server             : 'AGG_B'
Gate.ObjectServerB.CommonNames   : 'AGG_V'
```

Related property settings.

```
# AES_FIPS settings
ConfigCryptoAlg                  : 'AES_FIPS'
ConfigKeyFile                    : '$NCHOME/etc/security/crypt/objectserver.keyfile'
Gate.ObjectServerA.Username      : 'gateway'
Gate.ObjectServerB.Username      : 'gateway'
Gate.ObjectServerA.Password      : '@44:PMrAqIxi0tvExC8Gx3ELAQWc4luIOSmSso+I0mIO1Kk=@'
Gate.ObjectServerB.Password      : '@44:PMrAqIxi0tvExC8Gx3ELAQWc4luIOSmSso+I0mIO1Kk=@'
Gate.UsePamAuth                  : TRUE
```

1.9 Probe settings

The probes connecting to the aggregation object server uses the primary and backup object server name with the certificates Common Name.

```
Server                          : 'AGG_P'
ServerBackup                    : 'AGG_B'

# Object Servers Common Name values
SSLServerCommonName             : 'AGG_V'
```

The AES FIPS probe property settings are required when the $NCHOME/etc/security/fips.conf file is used on the probe server.

2 Object Server configuration

2.1 SSL/TLS certificate creation

The latest FAQ for creating TLSv1.2 certificates suitable for FIPS includes a set of scripts to assist in managing the SSL certificates.

Creating a TLSv1.2 compliant omni.kdb in Netcool/OMNIbus 8.1
https://www.ibm.com/support/pages/node/6557220

These use the additional nc_gskcmd option.

-sig_alg SHA512_WITH_RSA

The password for the omni.kdb needs to adhere to the FIPS requirements, when FIPS is enabled [$NCHOME/etc/fips.conf].
For example, the minimum password length is 14 characters.

The scripts create a local CA and Object Server SSL certificate in the $NCHOME/etc/security/keys directory.

Install the scripts into a suitable directory, for example $NCHOME/etc/security/scripts.

```
./create_CA_plus_os.sh

Usage: create_CA_plus_os.sh [OBJECT SERVER NAME]

Password requirements are:
 o The minimum length is 14 characters
 o At least one lower case character, one upper case character,
   and one digit or special characters
 o Each character must not occur more than three times
 o No more than two consecutive identical characters
 o All characters must be standard ASCII printable character set
   within the range from 0x20 to 0x7E inclusive

Default password is set to OMNIbus_Passw0rd

./import_certificates.sh

Usage: import_certificates.sh [OBJECT SERVER NAME]

Imports local Object Server arm files: NCOMS.arm and NCOMS_CA.arm
Creates /opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb if needed
```

2.2 AGG_P

On the primary aggregation object server, create the certificates for the Virtual Object Server, AGG_V.
First decide on a suitable password for the omni.kdb, with the required characters and longer than the minimum length.

Update the script with the required password.

```
JKSPASSWORD=OMNIbus_Passw0rd
```

Create the $NCHOME/etc/security/keys directory using the create_CA_plus_os.sh script.

```
./create_CA_plus_os.sh AGG_V
Created omni.kdb
Certificates found
* default, - personal, ! trusted, # secret key
-       AGG_V_CA
$NCHOME/etc/security/keys/AGG_V_CA.arm
Continue ?

Request AGG_V
Signed AGG_V
Received AGG_V
Certificates found
* default, - personal, ! trusted, # secret key
-       AGG_V_CA
-       AGG_V
$NCHOME/etc/security/keys/AGG_V.arm
$NCHOME/etc/security/keys/AGG_V_CA.arm
$NCHOME/etc/security/keys/AGG_V_req.arm
omni.kdb password is OMNIbus_Passw0rd
```

2.2.1 Object Server property file

File : $NCHOME/omnibus/etc/AGG_P.props

```
# SSL
Ipc.SSLCertificateLabel : 'AGG_V'
# AES FIPS
PasswordEncryption      : 'AES'
# End of File
```

Creating the key file for AES_FIPS encryption.

```
mkdir $NCHOME/etc/security/crypt
$NCHOME/omnibus/bin/nco_keygen -o $NCHOME/etc/security/crypt/objectserver.keyfile -l
256
```

Adding the key file settings to the property file.

```
ConfigCryptoAlg: 'AES_FIPS'
ConfigKeyFile  : '$NCHOME/etc/security/crypt/objectserver.keyfile'
```

Encrypting passwords and other strings.

```
$NCHOME/omnibus/bin/nco_aes_crypt -c AES_FIPS -k
$NCHOME/etc/security/crypt/objectserver.keyfile password
@44:1IffU8iSzJdW3Vq35Gj/wmqpPcI0zaN+flmtykEvRqI=@
```

Setting the encrypted password property.

```
PA.Name     : 'NCO_PA'
PA.Username: 'root'
PA.Password: '@44:1IffU8iSzJdW3Vq35Gj/wmqpPcI0zaN+flmtykEvRqI=@'
```

Adding the object server to the interfaces file

```
File : $NCHOME/etc/omni.dat

[AGG_P]
{
        Primary: primary_host ssl 4101
}
[AGG_V]
{
        Primary:  primary_host ssl 4101
        Backup: backup_host ssl 4101
}
```

Run nco_igen to create the interfaces files.

Start the object server.

nco_objserv -name AGG_P

Check the SSL certificate was attached using openssl.

```
openssl s_client -connect primary_host:4101
CONNECTED(00000003)
depth=1 C = GB, ST = London, L = SouthBank, O = IBM, OU = Support, CN = AGG_V_CA
---
Certificate chain
 0 s:/C=GB/ST=London/L=SouthBank/O=IBM/OU=Support/CN=AGG_V
   i:/C=GB/ST=London/L=SouthBank/O=IBM/OU=Support/CN=AGG_V_CA
 1 s:/C=GB/ST=London/L=SouthBank/O=IBM/OU=Support/CN=AGG_V_CA
   i:/C=GB/ST=London/L=SouthBank/O=IBM/OU=Support/CN=AGG_V_CA
…
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : AES128-SHA
```

Check nco_sql access
Login to the object server using Virtual Object Server name.

```
nco_sql -server AGG_V -user root
Password:

select Value from catalog.properties where PropName = 'Name';
go

Value
AGG_P
```

2.3 AGG_B

The Backup object server can use the same 'keys' directory as the Primary object server, to reduce administration and improve failover recovery.

On the Primary object server host, primary_host.

primary_host:
```
cd $NCHOME/etc/security
tar cvf AGG_V.keys.tar keys
```

Copy the AGG_V.keys.tar file to the backup_host, and untar in the $NCHOME/etc/security directory.

backup_host:
```
cd $NCHOME/etc/security
mv keys keys.default
tar xvf /tmp/AGG_V.keys.tar
```

Check the omni.kdb contents is installed correctly.

```
$NCHOME/bin/nc_gskcmd -cert -list -db "$NCHOME/etc/security/keys/omni.kdb" -stashed
Certificates found
* default, - personal, ! trusted, # secret key
-       AGG_V_CA
-       AGG_V
```

2.3.1 Object Server property file

File : $NCHOME/omnibus/etc/AGG_B.props

```
# SSL
Ipc.SSLCertificateLabel : 'AGG_V'
# AES FIPS
PasswordEncryption      : 'AES'
# End of File
```

Creating the key file for AES_FIPS encryption.

```
mkdir $NCHOME/etc/security/crypt
$NCHOME/omnibus/bin/nco_keygen -o $NCHOME/etc/security/crypt/objectserver.keyfile -l
256
```

Adding the key file settings to the property file.

```
ConfigCryptoAlg: 'AES_FIPS'
ConfigKeyFile  : '$NCHOME/etc/security/crypt/objectserver.keyfile'
```

Encrypting passwords and other strings.

```
$NCHOME/omnibus/bin/nco_aes_crypt -c AES_FIPS -k
$NCHOME/etc/security/crypt/objectserver.keyfile password
@44:1IffU8iSzJdW3Vq35Gj/wmqpPcI0zaN+flmtykEvRqI=@
```

Setting the encrypted password property.

```
PA.Name    : 'NCO_PA'
PA.Username: 'root'
PA.Password: '@44:1IffU8iSzJdW3Vq35Gj/wmqpPcI0zaN+flmtykEvRqI=@'
```

Adding the object server to the interfaces file

```
File : $NCHOME/etc/omni.dat

[AGG_B]
{
        Primary: backup_host ssl 4101
}
#
# Invert Primary/Backup for nco_sql -server AGG_V access
#
[AGG_V]
{
        Primary:  backup_host ssl 4101
        Backup: primary_host ssl 4101
}
[AGG_P]
{
        Primary: primary_host ssl 4101
}
```

Run nco_igen to create the interfaces files.

Start the object server.

```
nco_objserv -name AGG_B
```

Check the SSL certificate was attached using openssl.

```
openssl s_client -connect backup_host:4101
CONNECTED(00000003)
depth=1 C = GB, ST = London, L = SouthBank, O = IBM, OU = Support, CN = AGG_V_CA
---
Certificate chain
 0 s:/C=GB/ST=London/L=SouthBank/O=IBM/OU=Support/CN=AGG_V
   i:/C=GB/ST=London/L=SouthBank/O=IBM/OU=Support/CN=AGG_V_CA
 1 s:/C=GB/ST=London/L=SouthBank/O=IBM/OU=Support/CN=AGG_V_CA
   i:/C=GB/ST=London/L=SouthBank/O=IBM/OU=Support/CN=AGG_V_CA
…
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : AES128-SHA
```

Check nco_sql access
Login to the object server using Virtual Object Server name.

```
nco_sql -server AGG_V -user root
Password:

select Value from catalog.properties where PropName = 'Name';
go

Value
AGG_B
```

3 Aggregation gateway configuration

The AGG_GATE gateway is already configured in the $NCHOME/omnibus/gates/AGG_GATE directory and configured in the $NCHOME/etc/omni.dat. There is no need to set the AGG_GATE interfaces to SSL, for security, it could be defined as 'localhost'.

File : $NCHOME/omnibus/gates/AGG_GATE/AGG_GATE.props

```
# AGG_GATE configuration
MessageLevel                            : 'info'
Name                                    : 'AGG_GATE'
MessageLog                              : '$OMNIHOME/log/AGG_GATE.log'
Ipc.Timeout                             : 600
Gate.CacheHashTblSize                   : 50021
Gate.MapFile                            : '$OMNIHOME/gates/AGG_GATE/AGG_GATE.map'
Gate.StartupCmdFile                     :
'$OMNIHOME/gates/AGG_GATE/objserv_bi.startup.cmd'
Gate.Mapper.ForwardHistoricDetails      : TRUE
Gate.Mapper.ForwardHistoricJournals     : TRUE
Gate.ObjectServerA.Server               : 'AGG_P'
Gate.ObjectServerA.BufferSize           : 50
Gate.ObjectServerA.Description          : 'failover_gate'
Gate.ObjectServerA.TblReplicateDefFile  :
'$OMNIHOME/gates/AGG_GATE/AGG_GATE.tblrep.def'
Gate.ObjectServerA.UseBulkInsCmd        : TRUE
Gate.ObjectServerB.Server               : 'AGG_B'
Gate.ObjectServerB.BufferSize           : 50
Gate.ObjectServerB.Description          : 'failover_gate'
Gate.ObjectServerB.TblReplicateDefFile  :
'$OMNIHOME/gates/AGG_GATE/AGG_GATE.tblrep.def'
Gate.ObjectServerB.UseBulkInsCmd        : TRUE
# Synchronisation
Gate.Resync.Enable                      : TRUE
Gate.Resync.Type                        : 'TWOWAYUPDATE'
# OR
#Gate.Resync.Type                        : 'UPDATE'
Gate.Resync.LockType                    : 'PARTIAL'
#
# SSL configuration
#
Gate.ObjectServerA.CommonNames          : 'AGG_V'
Gate.ObjectServerB.CommonNames          : 'AGG_V'
#
# AES_FIPS
#
ConfigCryptoAlg                         : 'AES_FIPS'
ConfigKeyFile                           :
'$NCHOME/etc/security/crypt/objectserver.keyfile'
Gate.ObjectServerA.Username             : 'gateway'
Gate.ObjectServerB.Username             : 'gateway'
Gate.ObjectServerA.Password             : '@44:...+I0mIO1Kk=@'
Gate.ObjectServerB.Password             : '@44:...+I0mIO1Kk=@'
#
# Enable nco_sql login - Requiies PAM configuration
# cd /etc/pam.d
# cp login nco_g_objserv_bi
Gate.UsePamAuth                         : TRUE
# EOF
```

4 Collection layer configuration

On the collection object server, create the certificates for the local object server.
First decide on a suitable password for the omni.kdb, with the required characters and longer than the minimum length.

Update the script with the required password.

```
JKSPASSWORD=OMNIbus_Passw0rd
```

Create the $NCHOME/etc/security/keys directory using the create_CA_plus_os.sh script.

```
./create_CA_plus_os.sh COL_P_1
Created omni.kdb
Certificates found
* default, - personal, ! trusted, # secret key
-       COL_P_1_CA
$NCHOME/etc/security/keys/COL_P_1_CA.arm
Continue ?

Request COL_P_1
Signed COL_P_1
Received COL_P_1
Certificates found
* default, - personal, ! trusted, # secret key
-       COL_P_1_CA
-       COL_P_1
$NCHOME/etc/security/keys/COL_P_1.arm
$NCHOME/etc/security/keys/COL_P_1_CA.arm
$NCHOME/etc/security/keys/COL_P_1_req.arm
omni.kdb password is OMNIbus_Passw0rd
```

4.1.1 Object Server property file

File : $NCHOME/omnibus/etc/AGG_P.props

```
# SSL -
# There's no need to set the label as the object server name
# is the same as the label
# Ipc.SSLCertificateLabel : 'COL_P_1'
#
# AES FIPS
#
PasswordEncryption       : 'AES'
# End of File
```

Creating the key file for AES_FIPS encryption.

```
mkdir $NCHOME/etc/security/crypt
$NCHOME/omnibus/bin/nco_keygen -o $NCHOME/etc/security/crypt/objectserver.keyfile -l
256
```

Adding the key file settings to the property file.

```
ConfigCryptoAlg: 'AES_FIPS'
ConfigKeyFile  : '$NCHOME/etc/security/crypt/objectserver.keyfile'
```

Encrypting passwords and other strings.

```
$NCHOME/omnibus/bin/nco_aes_crypt -c AES_FIPS -k
$NCHOME/etc/security/crypt/objectserver.keyfile password
@44:1IffU8iSzJdW3Vq35Gj/wmqpPcI0zaN+flmtykEvRqI=@
```

Setting the encrypted password property.

```
PA.Name    : 'NCO_PA'
PA.Username: 'root'
PA.Password: '@44:1IffU8iSzJdW3Vq35Gj/wmqpPcI0zaN+flmtykEvRqI=@'
```

Adding the object server to the interfaces file

```
File : $NCHOME/etc/omni.dat

[COL_P_1]
{
        Primary: collection_host ssl 4101
}
[AGG_V]
{
        Primary:  primary_host ssl 4101
        Backup: backup_host ssl 4101
}
```

Run nco_igen to create the interfaces files.

Start the object server.

```
nco_objserv -name COL_P_1
```

Check the SSL certificate was attached using openssl.

```
openssl s_client -connect collection_host:4101
```

check nco_ping

```
nco_ping COL_P_1
NCO_PING: Server available.
```

Check nco_sql access

Login to the collection layer object server.

```
nco_sql -server COL_P_1 -user root
Password:

select Value from catalog.properties where PropName = 'Name';
go

Value
COL_P_1
```

Login to the aggregation object server using Virtual Object Server name.

```
nco_sql -server AGG_V -user root
Password:

select Value from catalog.properties where PropName = 'Name';
go

Value
AGG_V
```

4.2 Collection gateway configuration

The collection object server gateway requires connectivity to the Aggregation Object Server's, which requires the omni.kdb to be updated with the AGG_V certificates.

Update the omni.kdb with the AGG_V certificates.

On the primary_host
```
cd $NCHOME/etc/security
tar cvf AGG_V.certs.tar keys/AGG_V.arm keys/AGG_V_CA.arm
keys/AGG_V.arm
keys/AGG_V_CA.arm
```

Copy the tar file to the collection object server host.

On the collection_host
```
cd $NCHOME/etc/security
tar xvf AGG_V.certs.tar
keys/AGG_V.arm
keys/AGG_V_CA.arm
```

Add the certificates to the omni.kdb.

```
$NCHOME/bin/nc_gskcmd -cert -add \
-db "$NCHOME/etc/security/keys/omni.kdb" -stashed \
-trust enable \
-label "AGG_V_CA" \
-file "$NCHOME/etc/security/keys/AGG_V_CA.arm"

$NCHOME/bin/nc_gskcmd -cert -add \
-db "$NCHOME/etc/security/keys/omni.kdb" -stashed \
-trust enable \
-label "AGG_V" \
-file "$NCHOME/etc/security/keys/AGG_V.arm"


$NCHOME/bin/nc_gskcmd -cert -list \
-db "$NCHOME/etc/security/keys/omni.kdb" -stashed

Certificates found
* default, - personal, ! trusted, # secret key
!       AGG_V_CA
!       AGG_V
-       COL_P_1_CA
-       COL_P_1
```

Check connectivity to the primary and backup object server using openssl.

```
openssl s_client -connect primary_host:4101
openssl s_client -connect backup_host:4101
```

check nco_ping

```
nco_ping AGG_V
NCO_PING: Server available.
```

Check nco_sql access
Login to the collection layer object server.

```
nco_sql -server AGG_V -user root
Password:

select Value from catalog.properties where PropName = 'Name';
go

Value
AGG_P
```

4.3 Collection gateway property file

File : $NCHOME/omnibus/gates/C_TO_A_GATE_P_1/C_TO_A_GATE_P_1.props

```
#
# SSL -
# Because AGG_V is being used there is no need for CommonNames
#
# Gate.Reader.CommonNames: ''
# Gate.Writer.CommonNames: ''
#
# AES_FIPS
#
ConfigCryptoAlg                         : 'AES_FIPS'
ConfigKeyFile                           :
'$NCHOME/etc/security/crypt/objectserver.keyfile'
Gate.ObjectServerA.Username             : 'gateway'
Gate.ObjectServerB.Username             : 'gateway'
Gate.ObjectServerA.Password             : '@44:...+I0mIO1Kk=@'
Gate.ObjectServerB.Password             : '@44:...+I0mIO1Kk=@'
#
# Enable nco_sql login - Requiies PAM configuration
# cd /etc/pam.d
# cp login nco_g_objserv_bi
Gate.UsePamAuth                         : TRUE
#

# EOF
```

Where the password is set using key file.

```
$NCHOME/omnibus/bin/nco_aes_crypt -c AES_FIPS -k
$NCHOME/etc/security/crypt/objectserver.keyfile password
@44:1IffU8iSzJdW3Vq35Gj/wmqpPcI0zaN+flmtykEvRqI=@
```

5 Probe server configuration

The probes may be installed on the collection layer object servers or on specific hosts.

5.1 Example Probe property file for AGG_V

For probes connecting to the aggregation layer, only the AGG_V and AGG_V_CA certificates need to be added to the probe servers omni.kdb.

```
#
# SSL settings for object server
Server                          : 'AGG_P'
ServerBackup                    : 'AGG_B'
SSLServerCommonName             : 'AGG_V'
#
# AES_FIPS
ConfigCryptoAlg                 : "AES_FIPS"
ConfigKeyFile                   : "$NCHOME/etc/security/crypt/probes.key"
AuthUserName                    : "probe"
#AuthPassword                   : "NetcoolN3tc00l"
AuthPassword                    : "@44:zTbwd57SNosSxOnhxG75tUyriePbMybKobpkSJjaSr4=@"
#
# Best practice
NetworkTimeout                  : 15
PollServer                      : 60
ProbeWatchHeartbeatInterval     : 60
```

5.2 Example Probe property file for the collection layer

For the Collection layer the COL_P_1 and COL_B_1 certificates need to be added to the probe servers omni.kdb.

```
#
# SSL settings for object server
Server                          : 'COL_P_1'
ServerBackup                    : 'COL_B_1'
#
# AES_FIPS
ConfigCryptoAlg                 : "AES_FIPS"
ConfigKeyFile                   : "$NCHOME/etc/security/crypt/probes.key"
AuthUserName                    : "probe"
AuthPassword                    : "@44:zTbwd57SNosSxOnhxG75tUyriePbMybKobpkSJjaSr4=@"
#
# Best practice
NetworkTimeout                  : 15
PollServer                      : 60
ProbeWatchHeartbeatInterval     : 60
```

5.3 Probe servers key file

Creating the probe specific key file for AES_FIPS encryption on the probe server [once only].

```
mkdir $NCHOME/etc/security/crypt
$NCHOME/omnibus/bin/nco_keygen -o $NCHOME/etc/security/crypt/probes.key -l 256
```

The password is set using the probe specific key file.

```
$NCHOME/omnibus/bin/nco_aes_crypt -c AES_FIPS -k
$NCHOME/etc/security/crypt/probes.key password

@44:zTbwd57SNosSxOnhxG75tUyriePbMybKobpkSJjaSr4=@
```

6 fips.conf logic in scripts

How the fips.conf file affects the Netcool/OMNIbus scripts.

```
Script : bin/nc_gskcmd

if [ -f "$NCHOME/etc/security/fips.conf" ]; then
        exec "${GSK8CAPICMD}" "$@" -fips
else
        exec "${GSK8CAPICMD}" "$@"
fi

Script : omnibus/bin/nco_sql
if [ ! -f $NCHOME/etc/security/fips.conf ]; then
#
# If we're running the 'secure' nco_ssql, then get a login token
#
if [ "$SECURE" = "y" ]; then
        if [ ! -x $ARCH_BIN_DIR/nco_get_login_token ]; then
                err "Secure SQL not installed for $ARCH"
        fi

        if [ -z "$TOKENPASSWDOPTION" ]; then
                LOGINTOKEN=`$ARCH_BIN_DIR/nco_get_login_token -server $OMNISERVER
-username "$OMNIUSER" -networktimeout $OMNITIMEOUT`
        else
                LOGINTOKEN=`$ARCH_BIN_DIR/nco_get_login_token -server $OMNISERVER
-username "$OMNIUSER" -password "$TOKENPASSWD" -networktimeout $OMNITIMEOUT`
        fi
        if [ -z "$LOGINTOKEN" ]; then
                err "Failed to get login token"
        fi
        OMNIUSER="token-login"
        OMNIPASSWD="-P $LOGINTOKEN"
fi
fi
```