IBM z/OS Connect EE V3.0

Customization - Security with AT-TLS

Wildfire
Technical Hands-On Workshops

IBM

IBM Z
Wildfire Team –
Washington System Center

Lab Version Date: May 29, 2020

Table of Contents

> **Important:** On the desktop there is a file named *Security CopyPaste.txt*. This file contains commands and other text used in this workshop. Locate that file and open it. Use the copy-and-paste function (**Ctrl-C** and **Ctrl-V**) to enter commands or text. It will save time and help avoid typo errors. As a reminder text that appears in this file will be highlighted in yellow.

General Exercise Information and Guidelines

✓ This exercise requires the completion of the *zCEE Basic Configuration* and *zCEE Basic Security Configurations* exercises before it can be performed.

✓ This exercise requires using z/OS user identities *FRED* and *USER1*. The password for these users will be provided by the lab instructions.

✓ There are examples of server.xml scattered through this exercise. Your server.xml may differ depending on which exercises have been previously performed. Be sure the red lines in these examples re either added or already present.

✓ The acronyms RACF (resource access control facility) and SAF (*system authorization facility*) are used in this exercise. RACF is the IBM security manager product whereas SAF is a generic term for any security manager product, e.g. ACF2 or Top Secret or RACF. An attempt has been to use SAF when referring to information appropriate for any SAF product and to use RACF when referring to specific RACF commands or examples.

✓ Any time you have any questions about the use of IBM z/OS Explorer, 3270 screens, features or tools, do not hesitate to ask the instructor for assistance.

✓ Text in **bold** and highlighted in **yellow** in this document should be available for copying and pasting in a file named *Security CopyPaste* file on the desktop.

✓ Please note that there may be minor differences between the screen shots in this exercise versus what you see when performing this exercise. These differences should not impact the completion of this exercise.

Overview

This exercise demonstrates the steps required to enable TLS security between an MVS batch region and a z/OS Connect EE (zCEE) server. TLS security between these two endpoints requires the use of AT-TLS (Application Transparent-TLS). This document is only intended to be an introduction to AT-TLS, not an all-encompassing description of what can be done by AT-TLS. Therefore, only a simple AT-TLS outbound policy will be used in this exercise. Hopefully performing this exercise will provide the foundation required to fully exploit AT-TLS in other scenarios.

- First the RACF resources, e.g. digital certificates, keyrings will be defined and configured.
- z/OSMF will then be used to configure a simple AT-TLS outbound policy.
- The exercise will begin by submitting an API requester batch application that invokes an API using HTTP. This is done to become familiar with the application.
- The AT-TLS policy will be activated and messages and traces etc. will be reviewed to confirm the policy has been installed.
- Finally, the client API requester batch application will be executed again showing how HTTPS is used to encrypt the message and provide an authentication identity.

Mitch Johnson (mitchj@us.ibm.com)

Creating RACF resources

In this section, the required RACF resources will be created. Note that the digital certificates used by the batch client application were created in a previous exercise. Review member ZCEERCF4 to see the commands used to create USER1's personal certificate and the certificate authority (CA) certificate used to sign the personal certificates used in this exercise.

___1. Browse data set *USER1.ZCEE30.CNTL*. You should see the members in that data set.

___2. Next browse member **ZCEERCF6**, you should see the RACF commands below. Submit the job for execution.

```
RACDCERT ID(USER1) ADDRING(Liberty.KeyRing)

RACDCERT ID(USER1) CONNECT(LABEL('USER1') +
        RING(Liberty.KeyRing) DEFAULT)

RACDCERT ID(USER1) CONNECT(CERTAUTH LABEL('Liberty CA') +
        RING(Liberty.KeyRing))

PERMIT IRR.DIGTCERT.LISTRING +
  CLASS(FACILITY) ID(USER1) ACCESS(READ)

PERMIT IRR.DIGTCERT.LIST +
  CLASS(FACILITY) ID(USER1) ACCESS(READ)

SETR RACLIST(DIGTCERT DIGTRING FACILITY) REFRESH
```
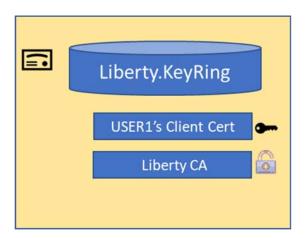
These commands

- Define a key ring for USER1
- Connect USER1's personal certificate (created earlier by job ZCEERCF4) to this key ring
- Connect the certificate authority (CA) public certificate used to sign USER1's personal certificate to this key ring.
- USER1 is given the required authority to access the key ring and certificate
- The in-storage profile for digital certificates resources are refreshed.

  N.B. The same CA was used to sign the server certificate that will be sent by the zCEE server this is the only CA certificate required on this key ring.

Mitch Johnson (mitchj@us.ibm.com)

Below is visual representation of the key ring just created

> **Tech-Tip:** In this example the digital certificates had been already present in the RACF data base. But if they had been provided by an external CA authority and stored in MVS data sets, they could have been added to RACF with these commands:
>
> ```
> racdcert id(USER1) withlabel('USER1)  add('USER1.USER1.P12') password('secret')
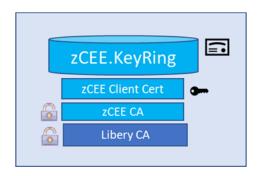> racdcert CERTAUTH withlabel('Liberty CA') add('USER1.LIBERTY.PEM')
> ```

Mitch Johnson (mitchj@us.ibm.com)

___3. Next browse member **ZCEESRCF7**. You should see the RACF commands below. Submit the job for execution if this job has not been previously submitted in another exercise.

```
/* Create personal certificate for zCEE outbound client request */
racdcert id(libserv) gencert subjectsdn(cn('zCEE Client Cert') +
ou('ATS') o('IBM')) withlabel('zCEE Client Cert') signwith(certauth +
label('zCEE CA')) notafter(date(2022/12/31))

 /* Create zCEE outbound key ring and connect certificates */
racdcert id(libserv) addring(zCEE.KeyRing)

racdcert id(libserv) connect(ring(zCEE.KeyRing) +
          label('zCEE CA') certauth usage(certauth))

racdcert id(libserv) connect(ring(zCEE.KeyRing) +
          label('Liberty CA') certauth usage(certauth))

 /* Connect CA certificate to Liberty inbound key ring */
racdcert id(libserv) connect(ring(Liberty.KeyRing) +
          label('zCEE CA') certauth usage(certauth))

 /* Connect default personal certific1ate */
racdcert id(libserv) connect(ring(zCEE.KeyRing) +
          label('zCEE Client Cert') default)

racdcert id(libserv) listring(zCEE.KeyRing)
racdcert id(libserv) list

setr raclist(digtcert digtring) refresh

connect   libserv  group(zceeusrs)
connect   libserv  group(gminvoke)
```

These commands

- Define a personal certificate for the zCEE server for use during outbound handshakes
- Define a key ring to be used for outbound handshakes
- Connect the zCEE server personal certificate to this key ring
- Connect the CA public certificate used to sign the zCEE server's outbound personal certificate to this key ring.
- Connect the CA public certificate used to sign the API provider server's certificate to this key ring.
- Connects the CA public certificate used to sign the zCEE server's outbound personal certificate to the API provider's key ring
- User LIBSERV is given the required authority to access their key ring and certificate
- The in-storage profile for digital certificates resources are refreshed.
- User LIBSERV is connected to the groups that provide access to this zCEE instance.

Below is visual representation of the key ring just created

Mitch Johnson (mitchj@us.ibm.com)

___4. Edit the *server.xml* configuration file for the *myServer* server found in directory. */var/zosconnect/servers/myServer* and change the include for *keyringMutual.xml* to an include of *keyringOutboundMutua.xml* and add an included for *shared.xml*, see below:

> ==*<include location="/${server.config.dir}/includes/keyingOutboundMutual.xml"/>*==
>
> ==*<include location="/${server.config.dir}/includes/shared.xml"/>*==

This step may not be required if another exercise has already been performed this action.

```
<include location="/${server.config.dir}/includes/safSecurity.xml"/>
<include location="/${server.config.dir}/includes/ipic.xml"/>
<include location="/${server.config.dir}/includes/keyringOutboundMutual.xml"/>
<include location="/${server.config.dir}/includes/groupAccess.xml"/>
<include location="/${server.config.dir}/includes/shared.xml"/>
```

```
        <!-- Enable features -->
        <featureManager>
            <feature>transportSecurity-1.0</feature>
        </featureManager>

        <sslDefault sslRef="DefaultSSLSettings"
          outboundSSLRef="OutboundSSLSettings" />

        <ssl id="DefaultSSLSettings"
           keyStoreRef="CellDefaultKeyStore"
           trustStoreRef="CellDefaultKeyStore"
           clientAuthenticationSupported="true"
           clientAuthentication="true"/>

        <keyStore id="CellDefaultKeyStore"
           location="safkeyring:///Keyring.LIBERTY"
           password="password" type="JCERACFKS"
           fileBased="false" readOnly="true" />

        <ssl id="OutboundSSLSettings"
           keyStoreRef="OutboundKeyStore"
           trustStoreRef="OutboundKeyStore"/>

        <keyStore id="OutboundKeyStore"
           location="safkeyring:///zCEE.KeyRing"
           password="password" type="JCERACFKS"
           fileBased="false" readOnly="true" />
```

Mitch Johnson (mitchj@us.ibm.com)

___5. Stop and restart the server with MVS commands *P BAQSTRT* and *S BAQSTRT*
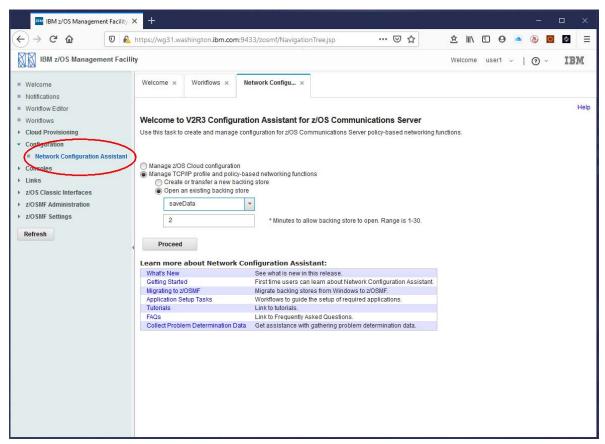
Configure the AT-TLS policy

z/OSMF will be used in this section to configure the AT-TLS configuration for the desired outbound policy.

___1. In a Firefox browser enter URL *https://wg31.washington.ibm.com:9433/zosmf* in the Firefox browser and you should see the *IBM z/OS Management Facility* window.

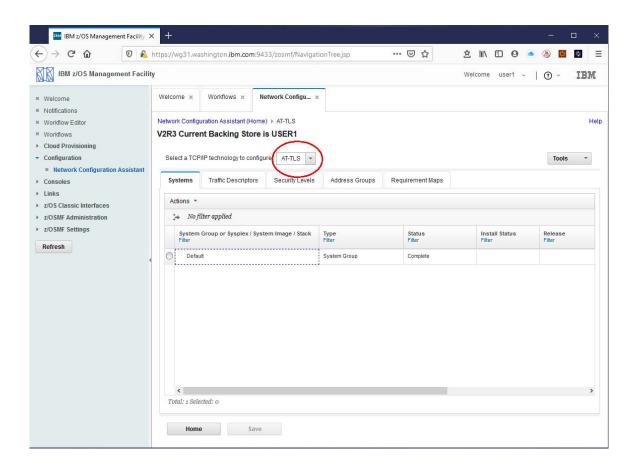___2. Enter *USER1* as the *z/OS USER ID* and USER1's password and click the **LOG IN** button.

___3. The *Welcome* screen should be displayed. On the left-hand side expand the *Configuration* tab to expose the *Network Configuration Assistance* option. Select this option to expose the *Network Configuration* tab.

___4. Select the radio button beside *Create or transfer a new backing store* option and click the **Proceed** button.

___5. On the next screen select the radio button beside *Create a New Backing Store File* and enter *USER1* in the area beside *File Name*. Press the **OK** button and press the **OK** button on the Information pop-up.

___6. On the *Network Configuration* tab use the pull-down arrow to select *AT-TLS* as the *TCP/IP technology to configure.*

___7. Select the radio button beside the *Default - System Group* and use the *Action* pull-down button to select *Add z/OS System Image* option.

Mitch Johnson (mitchj@us.ibm.com)

___8. On the *Add z/OS System Image* window enter **WG31** for the image *Name* and check the radio button beside *Simple name (as in an SAF product...)* and enter **Liberty.KeyRing** as the default AT-TLS key ring name. Click **OK** to continue.

> **Tech Tip:** The value for the key ring will be used if an explicit key ring is not provided for a policy.
>
> We recommend establishing a naming convention for key rings with each SAF identity using the same key ring name in the same context. Using this name as an example you could create a unique key ring named *Liberty.KeyRing* for SAF identities USER1, USER2, FRED, etc. Each user's key ring would have the same name but a different set of connected certificates. One default key ring specified at the image level covers all users.

___9. On the *Proceed to the Next Step?* pop-up click the **Proceed** button.

___10. The *Add TCP/IP Stack* screen should be displayed. Select this option to expose the *Network Configuration* tab. Enter **TCPIP1** as the name of the stack. Click **OK** to continue.

Tech-Tip: The value for the stack name was determine by the TCPIP Name display by entering the MVS command D TCPIP.

```
EZAOP50I TCPIP STATUS REPORT 007
COUNT    TCPIP NAME    VERSION     STATUS
-----    ----------    --------    --------------------------
    1    TCPIP1        CS V2R3     ACTIVE
*** END TCPIP STATUS REPORT ***
 EZAOP41I 'DISPLAY TCPIP' COMMAND COMPLETED SUCCESSFULLY
```

___11. Before any TCP/IP stack rules can be added, *Traffic Descriptors*, *Address Groups* and *Requirement Maps* need to be defined. Click **Cancel** on the *Proceed to the Next Step?* displayed at this time.

___12. This will display the window below:

Tech Tip: The Incomplete warning will be addressed shortly.

___13. Select the radio button beside *WG31* and use the *Actions* pull-down to select *Properties*. On the *Modify z/OS System Image* window select the *System Image Level Settings* tab and check all the trace level boxes as shown below. This is being done so we can confirm AT-TLS is being invoked and identify issues. Press **OK** to continue.

Mitch Johnson (mitchj@us.ibm.com)

___14. Select the *Traffic Descriptors* tab and use the *Actions* pull-down to select *New*.

___15. On the *New Traffic Descriptor* window enter ***zCEEClient*** as the name and use the *Actions* pull-down and select *New* to start the definition of a new traffic descriptor type.

___16. On the *New Traffic Type – TCP* window select the radio button beside *Ephemeral ports* under *Local port*. Select the radio button *Single port* under *Remote port* and enter **9443** as the port number. Select the radio button beside *Outbound only* under *Indicate the TCP connection direction*. Enter **USER1GET** in the area under *Jobname* and finally select the radio button beside *Client* under *AT-TLS Handshake Role*. Click **OK** to continue.

> **Tech-Tip:** This traffic definition is triggered when job name *USER1GET* running on the local TCP/IP stack opens a temporary or ephemeral port and tries to connect to port 9443, e.g. *outbound. A further requirement could be to require that SAF identity associated with the job be a specific value.* If all the defined conditions are met, AT-TLS will act as a surrogate for the client during a TLS handshake. Note the *KeyRing* tab can be used to specify the name of the key ring to be used for this handshake. Otherwise the default is to use the same key ring name defined for the z/OS System image, e.g. Liberty.KeyRing.
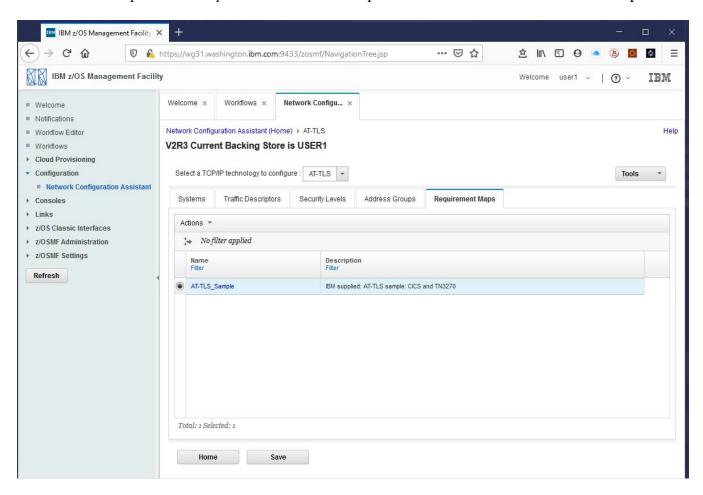
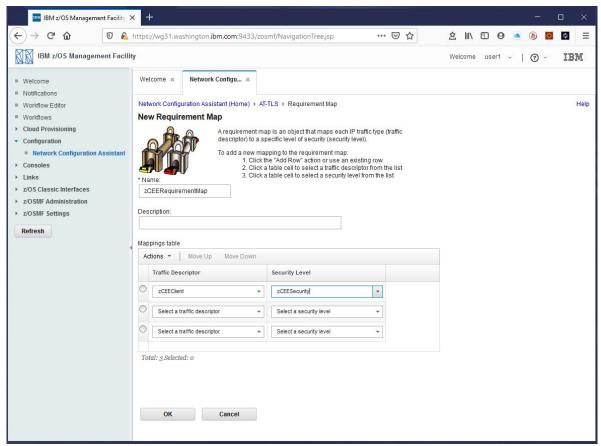___17. Click **OK** when the *New Traffic Descriptor* window is redisplayed.

___18. Next click the Security Levels tab and use the *Actions* pull-down button and to select the *New* option. On the *New Security Level* windows, enter ***zCEESecurity*** for the *Name* and check the box beside *TLS V1.2* and uncheck the other boxes. Click **Next** to display the *Cipher selection* options. Click **Next** to display the *Advanced Settings* options exploring as you like but there is no need to make any changes. Click Finish to continue.

Mitch Johnson (mitchj@us.ibm.com)

___19. Next click the *Requirement Maps* tab. Use the *Actions* pull-down button and to select the *New* option.

Mitch Johnson (mitchj@us.ibm.com)

___20. On the *New Requirement Map* window enter **zCEERequirementMap** as the *Name* and use the pull-down arrows to select *zCEEClient* as the *Traffic Descriptor* and *zCEESecurity* as the *Security Level* for this map. Click **OK** to continue.
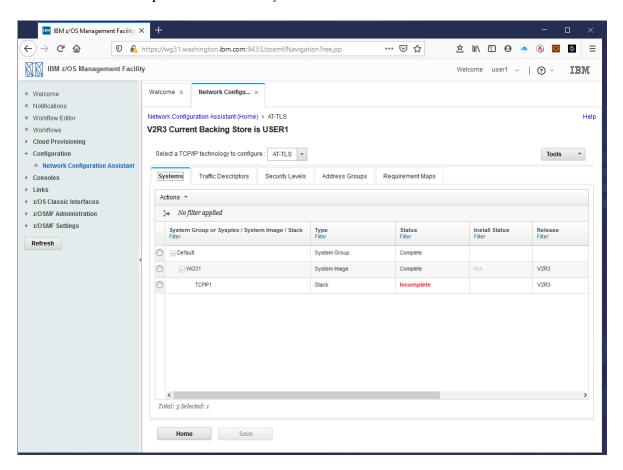
Mitch Johnson (mitchj@us.ibm.com)

___21. Select the radio button beside *zCEERequirementMap* and use the *Actions* pull-down to select the *View Details* options to display the window below. Review the details and click the **Close** button to continue.
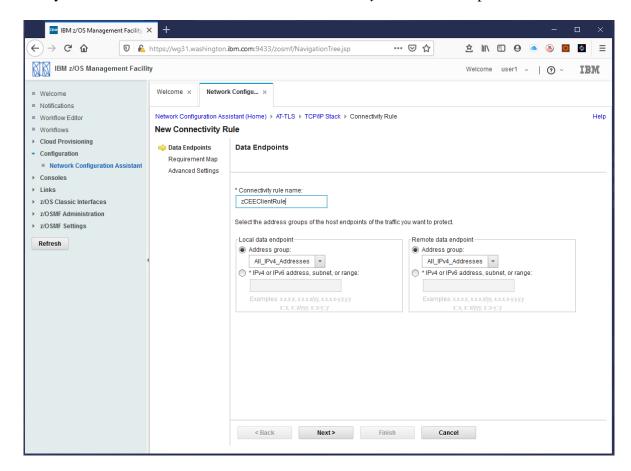

___22. Click the **Save** button to save the configuration.
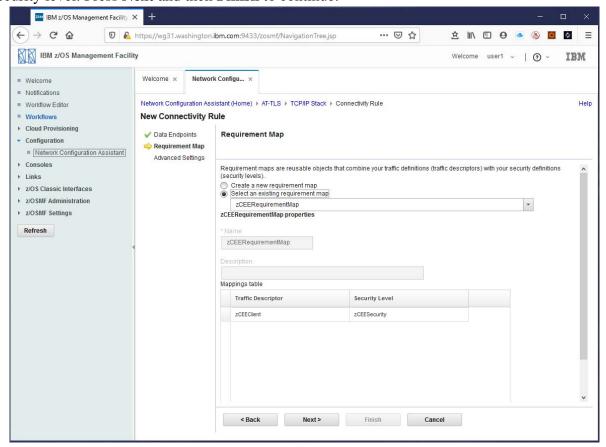
Mitch Johnson (mitchj@us.ibm.com)

___23. When the save has complete click on the *Systems* tab to return to this window.

___24. Select the radio button beside *TCPIP1* and use the *Actions* pull-down to select *Rules*. This is where these definitions will be tied together. Use the *Actions* pull-down and select *New* to create a new connectivity rule. Enter ***zCEEClientRule*** for the *Connectivity rule name* and press **Next** to continue.

___25. On the *New Connectivity Rule – Requirement Map* window select the radio button beside *Select an existing requirement map* and use the pull-down to select *zCEERequirementMap*. This should automatically populate the mapping table with *zCEEClient* as the traffic descriptor and *zCEESecurity* as the security level. Press **Next** and then **Finish** to continue.
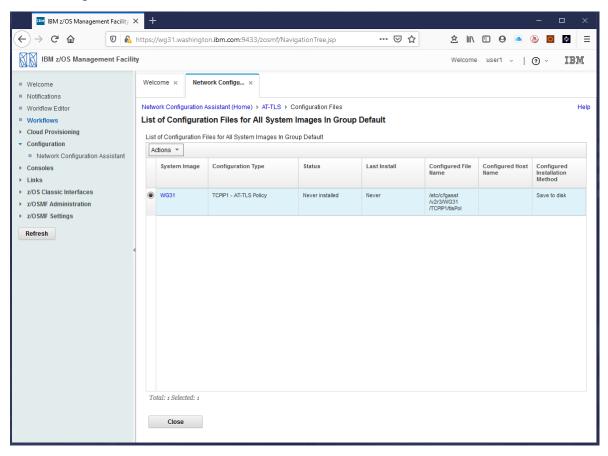
___26. Press **Close** to return to this window. Note that the status of the configuration is now complete.
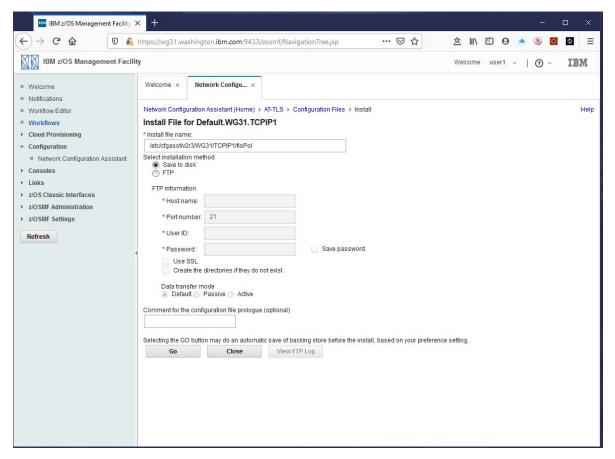

___27. Select the radio button beside *TCPIP1* and use the *Actions* pull-down to select *Install All Files for This Group.*
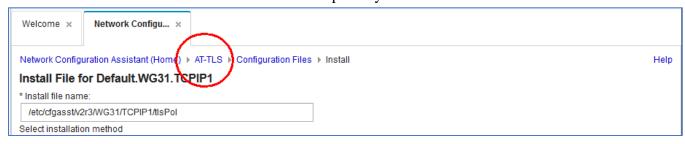
___28. On the *List of Configuration Files for All Systems Images in Group Default* window, select *WG31* and use the *Actions* pull-down to select *Install*.

Mitch Johnson (mitchj@us.ibm.com)

___29. On the *Install File for Default.WG31.TCPIP1* window click the **GO** button to continue.

___30. Click **OK** twice to continue.

___31. Next click on *AT-TLS* as shown below to return to the primary window.

___32. The AT-TLS configuration has been completed and is installed. But not active yet.

Mitch Johnson (mitchj@us.ibm.com)

Test the non-TLS connection from the batch job to the zCEE server

The JCL to execute the batch client can be found in *USER1.ZCEE30.CNTL*, the member name is *GETAPI*.

___1. But before any testing can be performed the server.xml file of the BAQSTRT may need to be updated depending on which previous exercises have been performed. Edit the *server.xml* in */var/zosconnect/servers/myServer* and ensure the following include files are present. You may need to add ==*<include location="${server.config.dir}/includes/apiRequesterHTTPS.xml" />*==.

```
<include location="/${server.config.dir}/includes/safSecurity.xml"/>
<include location="/${server.config.dir}/includes/ipicIDProp.xml"/>
<include location="/${server.config.dir}/includes/keyringOutboundMutual.xml"/>
<include location="/${server.config.dir}/includes/groupACCESS.xml"/>
<include location="/${server.config.dir}/includes/apiRequesterHTTPS.xml"/>
<include location="/${server.config.dir}/includes/shared.xml"/>
```

___2. To test a non-TTL connection, TLS security for the cscvinc_1.0.0 needs to be turned off temporarily. Edit file *apiRequesterHTTPS.xml* and add the *apiRequester* element as shown below. This disables the requirement for inbound HTTPS traffic for this specific API requester element.

```
<zosconnect_apiRequesters location="/var/zcee/shared/apiRequesters"
  idAssertion="ASSERT_ONLY">
  <apiRequester name="cscvinc_1.0.0" requireSecure="true" />
</zosconnect_apiRequesters>
```

___3. Stop and restart the server with MVS commands *P BASTRT* and *S BAQSTRT*.

___4. Submit the job in member **GETAPI** in *USER1.ZCEE30.CNTL*.

```
//GETAPI EXEC PGM=GETAPI,PARM='111111'
//STEPLIB  DD DISP=SHR,DSN=USER1.ZCEE30.LOADLIB
//         DD DISP=SHR,DSN=ZCEE30.SBAQLIB
//SYSPRINT DD SYSOUT=*
//CEEOPTS  DD *
 POSIX(ON),
 ENVAR("BAQURI=wg31.washington.ibm.com",
 "BAQPORT=9080")
```

___5. The job should complete with a condition code of 200 and have the following output for SYSPRINT.

```
NUMB:    111111
NAME:    C. BAKER
ADDRX:   OTTAWA, ONTARIO
PHONE:   51212003
DATEX:   26 11 81
AMOUNT: $0011.00
EIBRESP:    00000000
EIBRESP2:   00000000
USERID:     CICSUSER
HTTP CODE:  0000000200
```
___6.

Tech-Tip: An HTTP code of 200 indicates success. For an explanation of HTTP codes see URL https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

___4. Change the PARM='111111' to PARM='000000' and resubmit. This time the output should look something like this:

```
Error code: 0000000404
Error
msg:{"cscvincServiceOperationResponse":{"Container1":{"RESPONSE_CONTAINER
":{"CEIBRESP2":80,"FILEA_AREA":{"STAT":""
,"ADDRX":"","AMOUNT":"","PHONE":"","DATEX":"","NUMB":"000000","COMMENT":""
,"NAME":""},"ACTION":"S","USERID":"CICSUSER","
CEIBRESP":13}}}}
```

The CEIBRESP and CEIBRESP2 values are from a CICS program and the response codes received is returned when an EXEC CICS READ fails with a *Not Found* (404) condition.

The available records are listed below:

numb	name	addrx	Phone	datex	amount
000100	S. D. BORMAN	SURREY, ENGLAND	32156778	26 11 81	$0100.11
000102	J. T. CZAYKOWSI	WARWICK, ENGLAND	98356183	26 11 81	$1111.11
000104	M. B. DOMBEY	LONDON,ENGLAND	12846293	26 11 81	$0999.99
000106	A. I. HICKSON	CROYDON, ENGLAND	19485673	26 11 81	$0087.71
000111	ALAN TULIP	SARATOGA,CALIFORNIA	46120753	01 02 74	$0111.11
000762	SUSAN MALAIKA	SAN JOSE,CALIFORNIA	22312121	01 06 74	$0000.00
000983	J. S. TILLING	WASHINGTON, DC	34512120	21 04 75	$9999.99
001222	D.J.VOWLES	BOBLINGEN, GERMANY	70315551	10 04 73	$3349.99
001781	TINA J YOUNG	SINDELFINGEN,GERMANY	70319990	21 06 77	$0009.99
003210	B.A. WALKER	NICE, FRANCE	12345670	26 11 81	$3349.99
003214	PHIL CONWAY	SUNNYVALE, CAL.	34112120	00 06 73	$0009.99
003890	BRIAN HARDER	NICE FRANCE	00000000	28 05 74	$0009.99
004004	JANET FOUCHE	DUBLIN, IRELAND	71112121	02 11 73	$1259.99
004445	DR. P. JOHNSON	SOUTH BEND, S.DAK.	61212120	26 11 81	$0009.99
004878	ADRIAN JONES	SUNNYVALE, CALIF.	32212120	10 06 73	$5399.99
005005	A. E. DALTON	SAN FRANCISCO, CA.	00000001	01 08 73	$0009.99
005444	ROS READER	SARATOGA, CALIF.	67712120	20 10 74	$0809.99
005581	PETE ROBBINS	BOSTON, MASS.	41312120	11 04 74	$0259.99
006016	SIR MICHAEL ROBERTS	NEW DELHI, INDIA	70331211	21 05 74	$0009.88
006670	IAN HALL	NEW YORK, N.Y.	21212120	31 01 75	$3509.88
006968	J.A.L. STAINFORTH	WARWICK, ENGLAND	56713821	26 11 81	$0009.88
007007	ANDREW WHARMBY	STUTTGART, GERMANY	70311000	10 10 75	$5009.88
007248	M. J. AYRES	REDWOOD CITY, CALF.	33312121	11 10 75	$0009.88
007779	MRS. A. STEWART	SAN JOSE, CALIF.	41512120	03 01 75	$0009.88

Mitch Johnson (mitchj@us.ibm.com)

009000	P. E. HAVERCAN	WATERLOO, ONTARIO	09876543	21 01 75	$9000.00
100000	M. ADAMS	TORONTO, ONTARIO	03415121	26 11 81	$0010.00
111111	C. BAKER	OTTAWA, ONTARIO	51212003	26 11 81	$0011.00
200000	S. P. RUSSELL	GLASGOW, SCOTLAND	63738290	26 11 81	$0020.00
222222	DR E. GRIFFITHS	FRANKFURT, GERMANY	20034151	26 11 81	$0022.00
300000	V. J. HARRIS	NEW YORK, U.S.	64739801	26 11 81	$0030.00
333333	J.D. HENRY	CARDIFF, WALES	78493020	26 11 81	$0033.00
400000	C. HUNT	MILAN, ITALY	25363738	26 11 81	$0040.00
444444	D. JACOBS	CALGARY, ALBERTA	77889820	26 11 81	$0044.00
500000	P. KINGSLEY	MADRID, SPAIN	44454640	26 11 81	$0000.00
555555	S.J. LAZENBY	KINGSTON, N.Y.	39944420	26 11 81	$0005.00
600000	M.F. MASON	DUBLIN, IRELAND	12398780	26 11 81	$0010.00
666666	R. F. WALLER	LA HULPE, BRUSSELS	42983840	26 11 81	$0016.00
700000	M. BRANDON	DALLAS, TEXAS	57984320	26 11 81	$0002.00
777777	L.A. FARMER	WILLIAMSBURG, VIRG.	91876131	26 11 81	$0027.00
800000	P. LUPTON	WESTEND, LONDON	24233389	26 11 81	$0030.00
888888	P. MUNDY	NORTHAMPTON, ENG.	23691639	26 11 81	$0038.00
900000	D.S. RENSHAW	TAMPA, FLA.	35668120	26 11 81	$0040.00
999999	ANJI STEVENS	RALEIGH, N.Y.	84591639	26 11 81	$0049.00

Mitch Johnson (mitchj@us.ibm.com)

Activating the AT-TLS configuration

The AT-TLS configuration has been saved in an OMVS file but has not been installed in the active policy agent task (e.g. PAGENT).

___1. This instance of the policy agent has been configured to use the *SYSLOGD* daemon task to log messages

```
//PAGENT   EXEC PGM=PAGENT,REGION=0K,TIME=NOLIMIT,
//      PARM='ENVAR("_CEE_ENVFILE_S=DD:STDENV")/ -l SYSLOGD'
```

___2. The *SYSLOGD* daemon has been configured to write all log messages to file */var/syslogd/syslogall.log* (see the *syslog.conf* file in the **/etc** subdirectory).

```
##################################################################
#
# Write all messages with priority err and higher to log file errors.
#
#*.err             /var/log/%Y/%m/%d/errors
*.*               /var/syslogd/syslogall.log
#
```

___3. Use ISPF option 3.4 to access directory */var/syslogd* and the *v* line command to view *syslogall.log*. Go to the bottom of the file and you will something like what is shown below:

Mitch Johnson (mitchj@us.ibm.com)

___4. Start the policy agent task using MVS command *S PAGENT*.

___5. Exit the syslogall.log view session and reopen the file do a find for a subset of string ***EZZ8431I PAGENT STARTING*** and you should see these messages.

```
003414 0.0.0 port 22.
003415 main: EZZ8431I PAGENT STARTING
003416 main: Compiled on Sep 26 2016 at 18:37:59
003417 main: Use environment PAGENT_CONFIG_FILE = '/etc/pagent.conf'
003418 main: List all environment variables:
003419 main:   EXPORT '_CEE_ENVFILE_S=DD:STDENV'
003420 main:   EXPORT 'LIBPATH=/usr/lib:.'
003421 main:   EXPORT 'PAGENT_CONFIG_FILE=/etc/pagent.conf'
003422 main:   EXPORT 'PAGENT_LOG_FILE=SYSLOGD'
003423 main:   EXPORT 'TZ=EST5EDT'
003424 main:   EXPORT 'GSK_TRACE=0xFFFF'
003425 main: using code page 'IBM-1047'
003426 main: Using log level 511
```

___6. Do a find for the character string zCEE, e.g. *f zcee*, and you see multiple occurrences where the AT-TLS configuration elements added earlier are being processed.

```
003515 _profile: Processing Image TTLS config file: '/etc/cfgasst/v2r3/WG31/
003516  Processing:  'TTLSRule                      zCEEClientRule~1'
003517  Processing:  'TTLSGroupAction               gAct1~zCEEClient'
003518  Processing:  'TTLSEnvironmentAction         eAct1~zCEEClient'
003519  Processing:  'TTLSConnectionAction          cAct1~zCEEClient'
003520  Processing:  'TTLSConnectionAdvancedParms   cAdv1~zCEEClient'
003521  Processing:  'TTLSKeyringParms              keyR~WG31'
003522  Processing:  'IpAddrSet                     addr1'
003523  Processing:  'PortRange                     portR1'
003524  Processing:  'PortRange                     portR2'
003525 _profile: Finished processing Image TTLS config file
003526 Processing TTLS Group action 'gAct1~zCEEClient'
003527 Processing TTLS Connection action 'cAct1~zCEEClient'
003528 Processing TTLS Environment action 'eAct1~zCEEClient'
003529 ocessing TTLS rule 'zCEEClientRule~1'
```

___7. Go the bottom of this file and you see these messages

```
EZD1579I PAGENT POLICIES ARE NOT ENABLED FOR TCPIP1 : TTLS
EZZ8771I PAGENT CONFIG POLICY PROCESSING COMPLETE FOR TCPIP1 : QOS
EZZ8771I PAGENT CONFIG POLICY PROCESSING COMPLETE FOR TCPIP1 : TTLS
EZD1586I PAGENT HAS INSTALLED ALL LOCAL POLICIES FOR TCPIP1
Finished main config file update
```

Mitch Johnson (mitchj@us.ibm.com)

> **Tech-Tip:** If a policy or otherwise changed the new or updated policy can be installed with an MVS modify command,   *F PAGENT,REFRESH*

___8. The policy agent is active. The policies have been loaded, but there is one remaining step. The TCPIP stack has not been modified to enable TTLS.   AT-TLS has been configured on this image so is disabled by default and must be explicitly enabled.   This is done by using an MVS *VARY* command,

V TCPIP,,OBEY,SYS1.TCPPARMS(TTLS)

Where the contents of SYS1.TCPPARMS(TTLS) is simply: TCPCONFIG TTLS.

When you issue this command you should see these messages in the console log.

```
V TCPIP,,OBEY,SYS1.TCPPARMS(TTLS)
EZZ0060I PROCESSING COMMAND: VARY TCPIP,,OBEY,SYS1.TCPPARMS(TTLS)
EZZ0300I OPENED OBEYFILE FILE 'SYS1.TCPPARMS(TTLS)'
EZZ0309I PROFILE PROCESSING BEGINNING FOR 'SYS1.TCPPARMS(TTLS)'
EZZ0316I PROFILE PROCESSING COMPLETE FOR FILE 'SYS1.TCPPARMS(TTLS)'
EZZ0053I COMMAND VARY OBEY COMPLETED SUCCESSFULLY
```

> **Tech-Tip:** AT-TLS can be also be disabled with a VARY command,
> *V TCPIP,,OBEY,SYS1.TCPPARMS(NOTTLS)* where the contents of SYS1.TCPPARMS(NOTTLS) is
>
> TCPCONFIG NOTTLS

___9. To test a non-TTL connection, TLS security for the cscvinc_1.0.0 needs to be turned off temporary. Edit file *apiRequesterHTTPS.xml* and add the *apiRequester* element as shown below. This disables the requirement for inbound HTTPS traffic for this specific API requester element.

```
<zosconnect_apiRequesters location="/var/zcee/shared/apiRequesters"
  idAssertion="ASSERT_ONLY">
<apiRequester name="cscvinc_1.0.0" requireSecure="true" />
</zosconnect_apiRequesters>
```

> **Tech-Tip:** Use the *EA* (edit ASCII) to open the file in edit more.

___10. Stop and restart the server with MVS commands *P BASTRT* and *S BAQSTRT*.

___11. Edit member **GETAPI** in *USER1.ZCEE30.CNTL*, change the port to 9443 and be sure the PARM is set
to 111111. Submit the job for execution.

```
//GETAPI EXEC PGM=GETAPI,PARM='111111'
//STEPLIB  DD DISP=SHR,DSN=USER1.ZCEE30.LOADLIB
//         DD DISP=SHR,DSN=ZCEE30.SBAQLIB
//SYSPRINT DD SYSOUT=*
//CEEOPTS  DD *
 POSIX(ON),
 ENVAR("BAQURI=wg31.washington.ibm.com",
 "BAQPORT=9443")
```

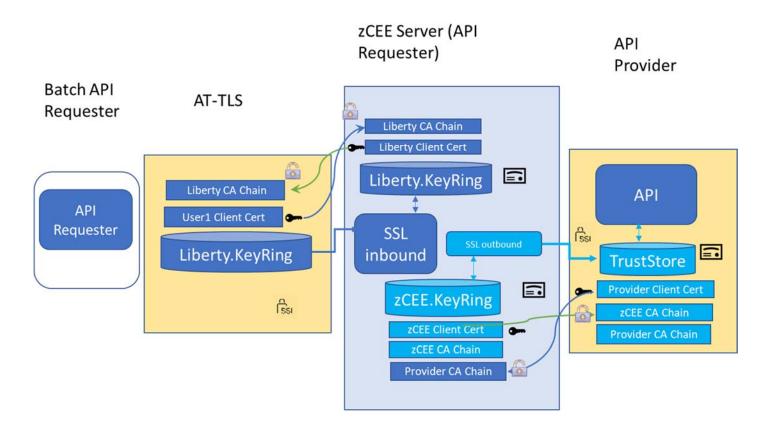___12. It should complete with a return code 200 with the record displayed as before.

```
NUMB:  111111
NAME:  C. BAKER
ADDRX:  OTTAWA, ONTARIO
PHONE: 51212003
DATEX: 26 11 81
AMOUNT: $0011.00
EIBRESP:   00000000
EIBRESP2:  00000000
USERID:    LIBSERV
HTTP CODE:  0000000200
```

Mitch Johnson (mitchj@us.ibm.com)

___13. *The results raise a question. Why is the user identity LIBSERV and not USER1*? For an explanation see
below.

The diagram below shows the identity that flows from the API requester.   The API requester identity
does not flow to the API provider. Since it is an TLS connection and mutual authentication being
enabled, the RACF identity mapped to the zCEE server client certificate (LIBSERV) is used for
authorization checks and, in this case, propagated to the API provider (a CICS application).

___14. Close and view the *syslogall.log* file again. At the bottom of the file you should see messages like these:

```
EZD1281I TTLS Map   CONNID: 00001E0C LOCAL: 192.168.17.201..7419 REMOTE:
192.168.17.201..9443 JOBNAME: USER1GET USERID: USER1 TYPE: OutBound STATUS:
Enabled RULE: zCEEClientRule~1 ACTIONS: gAct1~zCEEClient eAct1~zCEEClient
cAct1~zCEEClient
 EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000001 CONNID: 00001E0C  RC:
0 Initial Handshake 00000050115258D0 0000005011522750 TLSV1.2  F0F0F3F5
```

These messages are recording the role of AT-TLS in the handshake with the zCEE server.

___15. Change the jobname to something else and submit the job again for execution. This time the job should terminate with a return code of 5 and these messages in the SYSOUT

```
Error code: 0000000005
Error msg:BAQI0005E: Unable to send request to or receive response from the
z/OS Connect EE server.HWTHRQST RC=262(0x106) RSC=0 RSN=Connectivity appears
to be lost




Error origin:STUB
```

The *messages.log* file for the server will have this message.

```
CWWKO0801E: Unable to initialize SSL connection. Unauthorized access was
denied or security settings have expired. Exception is
javax.net.ssl.SSLException: Unrecognized SSL message, plaintext connection?
```

A connection attempt was made to a HTTPS port using HTTP. The AT-TLS policy that acts as a surrogate client was not trigger because AT-TLS will only be triggered when the job name matches the value in the traffic descriptor configuration element.

Summary

An AT-TLS policy has been created and used to encrypt traffic from a batch API requester client application. Be introducing intentional errors we have confirmed the TLS handshakes are taking place and/or failing as expected.