

IBM z/OS Connect EE V3.0

# Customization - Security and Db2



*Lab Version Date: December 11, 2020*

## Table of Contents

<b>Overview .....</b>	<b>4</b>
<b>Enabling RACF Pass Tickets.....</b>	<b>5</b>
<i>Test with RACF Pass Tickets not enabled .....</i>	<i>5</i>
<i>Test with identity propagation enabled using RACF Pass Tickets.....</i>	<i>8</i>
<i>Summary .....</i>	<i>10</i>
<b>Configuring TSL security to a Db2 Subsystem .....</b>	<b>11</b>
<i>Creating Db2 SAF resources .....</i>	<i>11</i>
<i>Configure the AT-TLS policy .....</i>	<i>17</i>
<b>Activating the AT-TLS configuration .....</b>	<b>39</b>
<i>Test the TLS connection from the zCEE Server to a Db2 subsystem.....</i>	<i>43</i>
<i>Optional .....</i>	<i>45</i>
<b>Summary .....</b>	<b>45</b>
<b>Appendix – AT-TLS Policy Agent Configuration File .....</b>	<b>46</b>

**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 are 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 security between a z/OS Connect server and a Db2 subsystem.

In part one of the exercise, the use of RACF pass tickets will be configured. RACF pass tickets can be used to pass the z/OS Connect authenticated RACF identity to a Db2 subsystems for Db2 authorization checks.

In part two of the exercise, TLS support will be added by configuring an AT-TLS policy. The presence of this policy will act as a surrogate for handing the server role on behalf of the Db2 subsystem.

## ***Enabling RACF Pass Tickets***

When sending a request from an z/OS Connect server to Db2 the identity used for Db2 authorization checks is either the identity configured in the basic authentication element for the Db2 client connection or the identity associated with the client certificate exchanged during a mutual authentication handshake. In both cases, this identity is associated with the server and not the individual identities under which the requests are running.

To provide identity assertion of the individual requests requires the use of RACF Pass Tickets. When RACF Pass Tickets are enabled, the z/OS Connect server will obtain a pass ticket from RACF and send this ticket (token) with the request to Db2. When the request arrives at the Db2 subsystem Db2 will validate this ticket with RACF and extract the z/OS Connect authenticated identity from the token and use for subsequent Db2 authorization checks.

### ***Test with RACF Pass Tickets not enabled***

But first, let's explore invoking an Db2 API and observe the behavior when identity propagation is not enabled.

- \_\_\_1. Submit the job in *DB2REST1* in data set *USER1.ZCEE30.CNTL* for execution. This will bind the *selectEmployee* Db2 REST service.

```
//BIND EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=DSN1210.DB2.SDSNEXIT,DISP=SHR
// DD DSN=DSN1210.DB2.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DSNSTMT DD *
    SELECT EMPNO AS "employeeNumber", FIRSTNME AS "firstName",
           MIDINIT AS "middleInitial", LASTNAME AS "lastName",
           WORKDEPT AS "department", PHONENO AS "phoneNumber",
           JOB AS "job"
    FROM USER1.EMPLOYEE WHERE EMPNO = :employeeNumber
//SYSTSIN DD *
DSN SYSTEM(DSN2)
BIND SERVICE(SYSIBMSERVICE) -
NAME("selectEmployee") -
SQLENCODING(1047) -
DESCRIPTION('Select an employee from table USER1.EMPLOYEE')
```

- \_\_\_2. Edit the *server.xml* configuration file for the *myServer* server, e.g. */var/zosconnect/servers/myServer/server.xml* and add includes for *shared.xml* and *db2.xml* (if not already present) see below:

```
<include location="${server.config.dir}/includes/db2.xml"/>
```

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

```
<include location="${server.config.dir}/includes/safSecurity.xml"/>
<include location="${server.config.dir}/includes/ipic.xml"/>
<include location="${server.config.dir}/includes/keyringMutual.xml"/>
<include location="${server.config.dir}/includes/groupAccess.xml"/>
<include location="${server.config.dir}/includes/db2.xml"/>
<include location="${server.config.dir}/includes/shared.xml"/>
```

This will install some predefined services and APIs in the server.

- \_\_\_3. Stop and restart the server with MVS commands *P BAQSTRT* and *S BAQSTRT*.

**Tech-Tip:** MVS and JES2 commands can be entered from SDSF by enter a / (slash) on the command line followed by the command itself (e.g. /D T). The command results can be found in the system log. If a command is especially long, then simply entering a / (slash) to display a *SDSF – System Command Extension* panel where a command can span multiple lines. When an MVS command must be entered, the instructions in these exercises will indicate that the command is an MVS command. MVS commands can be enter at the prompt by using the / (slash) prefix or using the *SDSF – System Command Extension* panel.

- \_\_\_4. Open a DOS command prompt and go to directory *c:/z/admin*.

- \_\_\_5. Enter the cURL command below:

```
curl -X get --cacert certauth.pem --cert fred.p12:secret --cert-type P12
https://wg31.washington.ibm.com:9443/db2/employee/000010
```

:

```
c:\z\admin>curl -X get --cacert certauth.pem --cert fred.p12:secret --cert-type P12
https://wg31.washington.ibm.com:9443/db2/employee/000010
{ "StatusDescription": "Execution Successful", "ResultSet
Output": [ { "firstName": "CHRISTINE", "lastName": "HAAS", "middleInitial": "I", "phoneNumber":
"3978", "department": "A00", "job": "PRES
", "employeeNumber": "000010" } ], "StatusCode": 200 }
```

\_\_\_6. Enter the cURL command below:

```
curl -X GET --cacert certauth.pem --cert user1.p12:secret --cert-type P12  
https://wg31.washington.ibm.com:9443/db2/employee/000010
```

You should see the same results because the request to Db2 are using the basic authentication as configured in the db2.xml (see below).

```
<zoscconnect_zosConnectServiceRestClientConnection id="Db2Conn"  
  host="wg31.washington.ibm.com"  
  port="2446"  
  basicAuthRef="dsn2Auth" />  
  
<zoscconnect_zosConnectServiceRestClientBasicAuth id="dsn2Auth"  
  userName="USER1"  
  password="USER1" />
```

USER1 has READ access to DSNR resource DSN2.REST.

## *Test with identity propagation enabled using RACF Pass Tickets*

Next enable the use of RACF pass tickets in the server.xml file and retest and the observe the results.

\_\_\_1. Begin by submitting the job in member *DB2PTKT* in data set *USER1.ZCEE30.CNTL*.

```
RDEFINE PTKTDATA DSN2APPL SSIGNON(KEYMASK(123456789ABCDEF0)) +
APPLDATA('NO REPLAY PROTECTION')

RDEFINE PTKTDATA IRRPTAUTH.DSN2APPL.* UACC(NONE)
PERMIT IRRPTAUTH.DSN2APPL.* ID(LIBSERV) CLASS(PTKTDATA) ACC(UPDATE)

SETROPTS RACLIST(PTKTDATA) REFRESH
```

These commands define the required *PTKTDATA* resource *DSN2APPL*.

**Tech-Tip:** The value DSN2APPL was derived from the Db2 LU name in the DSNL004I startup message, for example.

```
DSNL004I -DSN2 DDF START COMPLETE 906
LOCATION DSN2LOC
LU      USIBMWZ.DSN2APPL
GENERICLU -NONE
DOMAIN  WG31.WASHINGTON.IBM.COM
TCPPORT 2446
SECPORT 2445
RESPORT 2447
IPNAME  -NONE
OPTIONS:
PKGREL = COMMIT
```

The value for the key mask was an arbitrary 16 hexadecimal string. If multiple RACF databases are involved this value must be the same for all.



- \_\_\_2. Edit the server's server.xml file and change the include for *db2.xml* to an include for *db2passTicket.xml*.

```
<include location="{server.config.dir}/includes/db2passTicket.xml"/>
```

```
<include location="{server.config.dir}/includes/safSecurity.xml"/>
<include location="{server.config.dir}/includes/ipic.xml"/>
<include location="{server.config.dir}/includes/keyringMutual.xml"/>
<include location="{server.config.dir}/includes/groupAccess.xml"/>
<include location="{server.config.dir}/includes/db2passTicket.xml"/>
<include location="{server.config.dir}/includes/shared.xml"/>
```

The contents of db2passTicket are shown below.

```
<zoscconnect_zosConnectServiceRestClientConnection id="Db2Conn"
  host="wg31.washington.ibm.com"
  port="2446"
  basicAuthRef="dsn2Auth" />

<zoscconnect_zosConnectServiceRestClientBasicAuth id="dsn2Auth"
  appName="DSN2APPL" />
```

- \_\_\_3. Refresh the server's configuration using the MVS command **F BAQSTRT,REFRESH,CONFIG**

- \_\_\_4. Enter the cURL command below:

```
curl -X get --cacert certauth.pem --cert fred.p12:secret --cert-type P12
https://wg31.washington.ibm.com:9443/db2/employee/000010
```

```
c:\z\admin>curl -X get --cacert certauth.pem --cert fred.p12:secret --cert-type P12
https://wg31.washington.ibm.com:9443/db2/employee/000010
{"errorMessage":"BAQR0429W: API db2employee encountered an error while processing a
request under URL https://wg31.washington.ibm.com:9443/db2/employee/000010."}
```

- \_\_\_5. Review the JES log for the DDF task *DSN2DIST* and you should see this message.

```
ICH408I USER(FRED      ) GROUP(ATSGRP  ) NAME(USER FRED
DSN2.REST CL(DSNR      )
INSUFFICIENT ACCESS AUTHORITY
ACCESS INTENT(READ     ) ACCESS ALLOWED(NONE    )
```

Identity FRED does not have access to the DSNR REST resource protecting this Db2 subsystem.

\_\_\_6. Enter the cURL command below:

```
curl -X get --cacert certauth.pem --cert user1.p12:secret --cert-type P12  
https://wg31.washington.ibm.com:9443/db2/employee/000010
```

```
c:\z\admin>curl -X get --cacert certauth.pem --cert user1.p12:secret --cert-type P12  
https://wg31.washington.ibm.com:9443/db2/employee/000010  
{ "StatusDescription": "Execution Successful", "ResultSet  
Output": [ { "firstName": "CHRISTINE", "lastName": "HAAS", "middleInitial": "I", "phoneNumber":
```

Identity USER1 does access to the DSNR REST resource protecting this Db2 subsystem.

## Summary

In this section a simple REST client has been used to invoke an API which accesses a Db2 subsystem. The API was initially tested without RACF Pass Tickets and using the basic authentication configured in the server. Required RACF resources were then defined and changes were made to the server.xml so RACF Pass Tickets would be used between the server and Db2. Finally, the cURL command was used to demonstrate that the identity associated with the client certificate (fred.p12 and user1.p12) provided by the cURL command were propagated to Db2 for authorization checks.

## Configuring TLS security to a Db2 Subsystem

Adding TLS support to a Db2 subsystem requires the creation of a key ring belonging to the identity under which the Db2 subsystem is executing (look for message IEF695I in the DSN2DSNT task's JES messages). This key ring contains the personal and all the certificate authority certificates that will be used during TLS handshakes. The creation of the key ring and the connection of certificates to the key ring are done using the RACDCERT RACF commands.

### Creating Db2 SAF resources

- \_\_\_1. Browse data set *USER1.ZCEE30.CNTL*. You should see the members in that data set.
- \_\_\_2. Browse member **ZCEESSL** (SSL client role), you should see the RACF commands below. Submit the job for execution only 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 certificate */
racdcert id(libserv) connect(ring(zCEE.KeyRing) +
label('zCEE Client Cert') default)

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

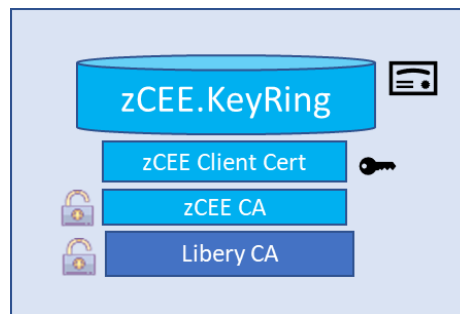
setr raclist(digtcert digtring) refresh

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 certificate authority (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 a visual representation of the key ring just created:



- \_\_\_3. Edit the *server.xml* configuration file for the *myServer* server, e.g. */var/zosconnect/servers/myServer/server.xml* and change the include for file *keyringMutual.xml* to an include of file *keyringOutboundMutua.xml*, see below:

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

\_\_\_4. Next browse member **DB2TLS**. The job contains the RACF commands below. Submit the job for execution.

```
/* Create a CA certificate for DB2 */
racdcert certauth gencert subjectsdn(cn('Db2 CA') ou('ATS') +
ou('ATS') o('IBM')) withlabel('DB2 CA') keyusage(certsign) +
notafter(date(2022/12/31))

/* Create a server certificate for DB2 client request */
racdcert id(DB2USER) gencert subjectsdn(cn('wg31.washington.ibm.com') +
ou('ATS') o('IBM')) withlabel('DB2USER') signwith(certauth +
label('DB2 CA')) notafter(date(2021/12/31))

setr raclist(digtcert,digtnmap) refresh

/* Create DB2 key ring and connect CA and personal certificates */
racdcert id(db2user) addring(Db2.KeyRing)

racdcert id(db2user) connect(ring(Db2.KeyRing) +
label('DB2 CA') certauth usage(certauth))

racdcert id(db2user) connect(ring(Db2.KeyRing) +
label('zCEE CA') certauth usage(certauth))

racdcert id(libserv) connect(ring(zCEE.KeyRing) +
label('DB2 CA') certauth usage(certauth))

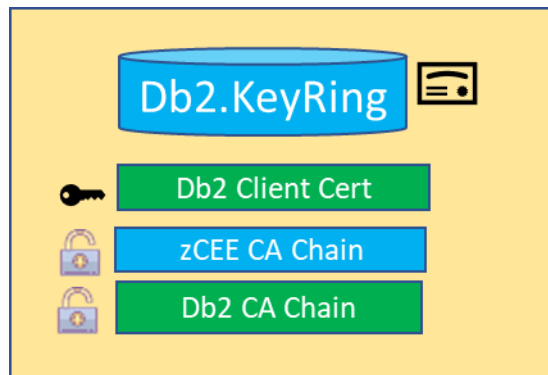
/* Connect default personal certificate */
racdcert id(db2user) connect(ring(Db2.KeyRing) +
label('DB2USER') default

setropts raclist(digtring,digtnmap) refresh
```

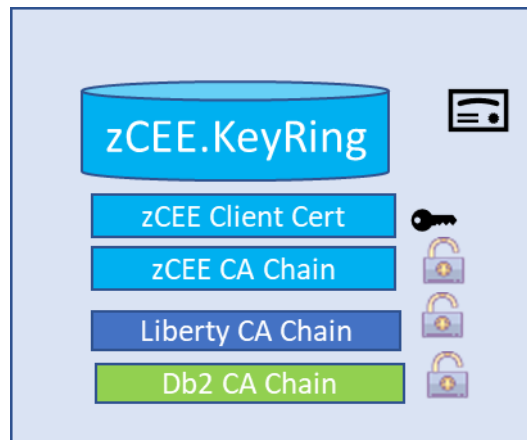
These commands

- Define a certificate authority certificate used to sign Db2 certificates used during TLS handshakes.
- Define a personal certificate for the Db2 server for use during TLS handshakes.
- Define a key ring to be used for TLS handshakes.
- Connect the Db2 server personal certificate to this key ring.
- Connect the CA public certificate used to sign the Db2 server's certificate to this key ring.
- Connect the CA public certificate used to sign the zCEE server's outbound personal certificate to this key ring.
- The in-storage profile for digital certificates resources are refreshed.

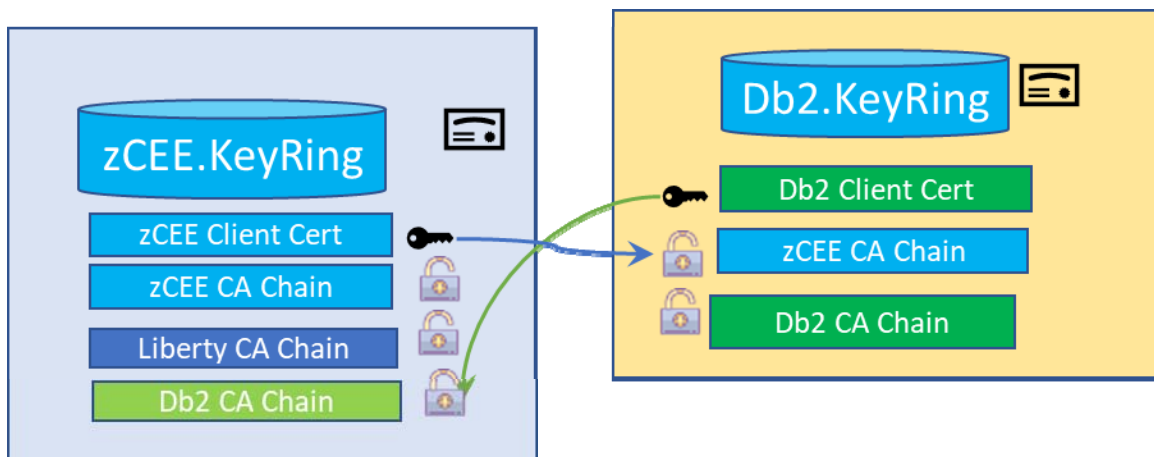
Below is a visual representation of the key ring just created:



And the update made to the zCEE server's outbound keyring:



And the handshakes will flow as shown below:



1. Edit the *server.xml* configuration file for the *myServer* server, e.g. */var/zosconnect/servers/myServer/server.xml* and change the include for file *keyringMutual.xml* to an include of file *keyringOutBoundMutual.xml*, see below:

```
<include location="$${server.config.dir}/includes/keyringOutboundMutual.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/imsDatabase.xml"/>
<include location="$${server.config.dir}/includes/db2passTicket.xml"/>
<include location="$${server.config.dir}/includes/shared.xml"/>
```

The contents of *keyringOutboundMutual.xml* are shown below.

```
<!-- 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" />
```

- \_\_\_2. Enter MVS commands *P BAQSRT and S BASSTR* to refresh the zCEE server's runtime configuration.

**Tech-Tip:** Updates to keyrings could have been refreshed in the server by using this command :

*F BAQSTR,REFRESH,KEYSTORE*

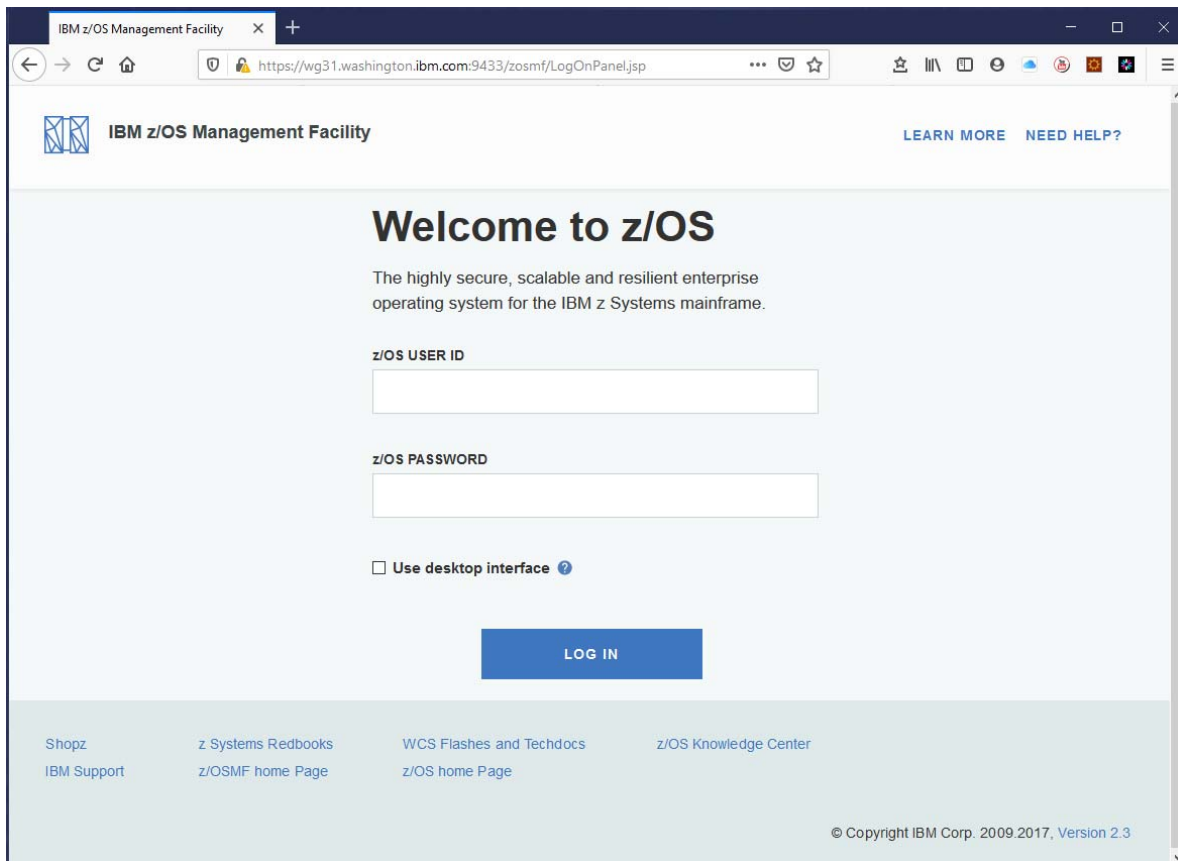
This would have dynamically made the information about CICS CA certificate available in the zCEE runtime.



## Configure the AT-TLS policy

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

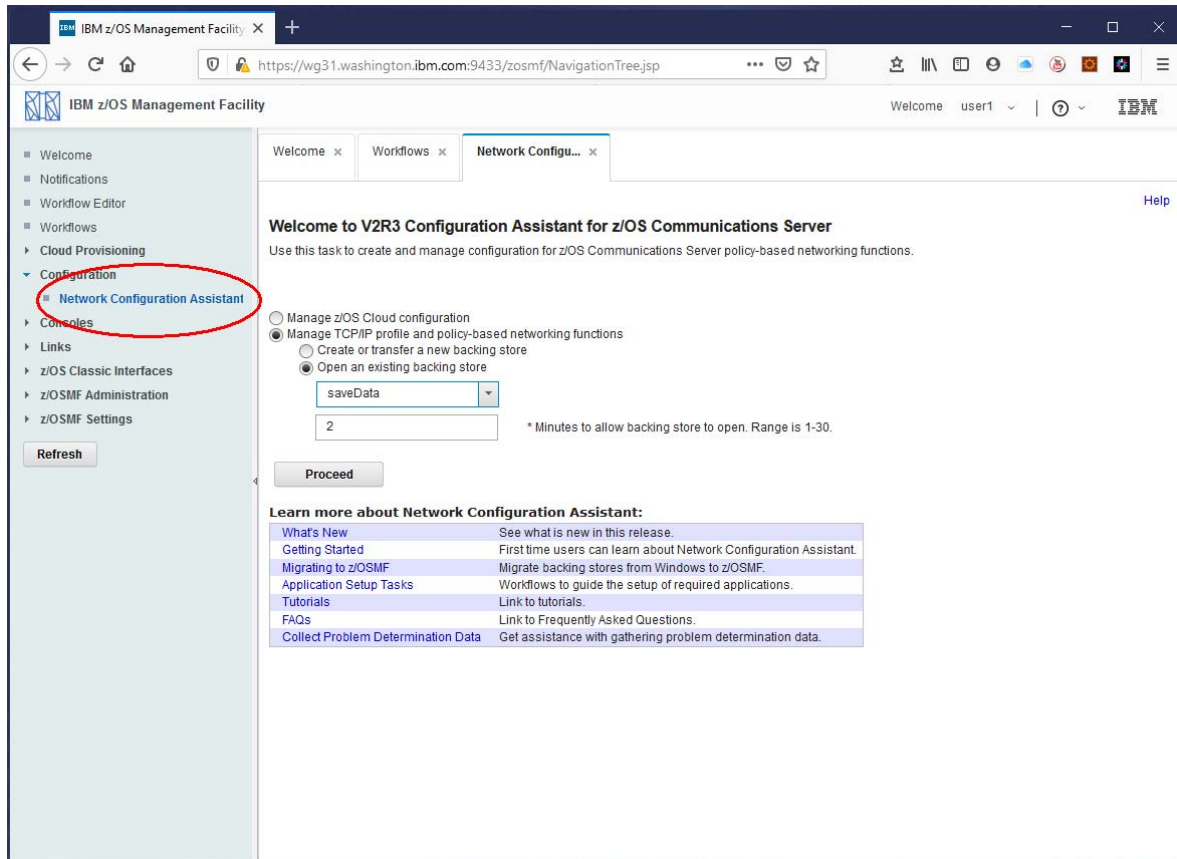
1. In a Firefox browser enter URL <https://wg31.washington.ibm.com:9433/zosmf> and you should see the *IBM z/OS Management Facility* window.



Note that some of the AT-TLS configurations steps described here may have been performed in another exercise.

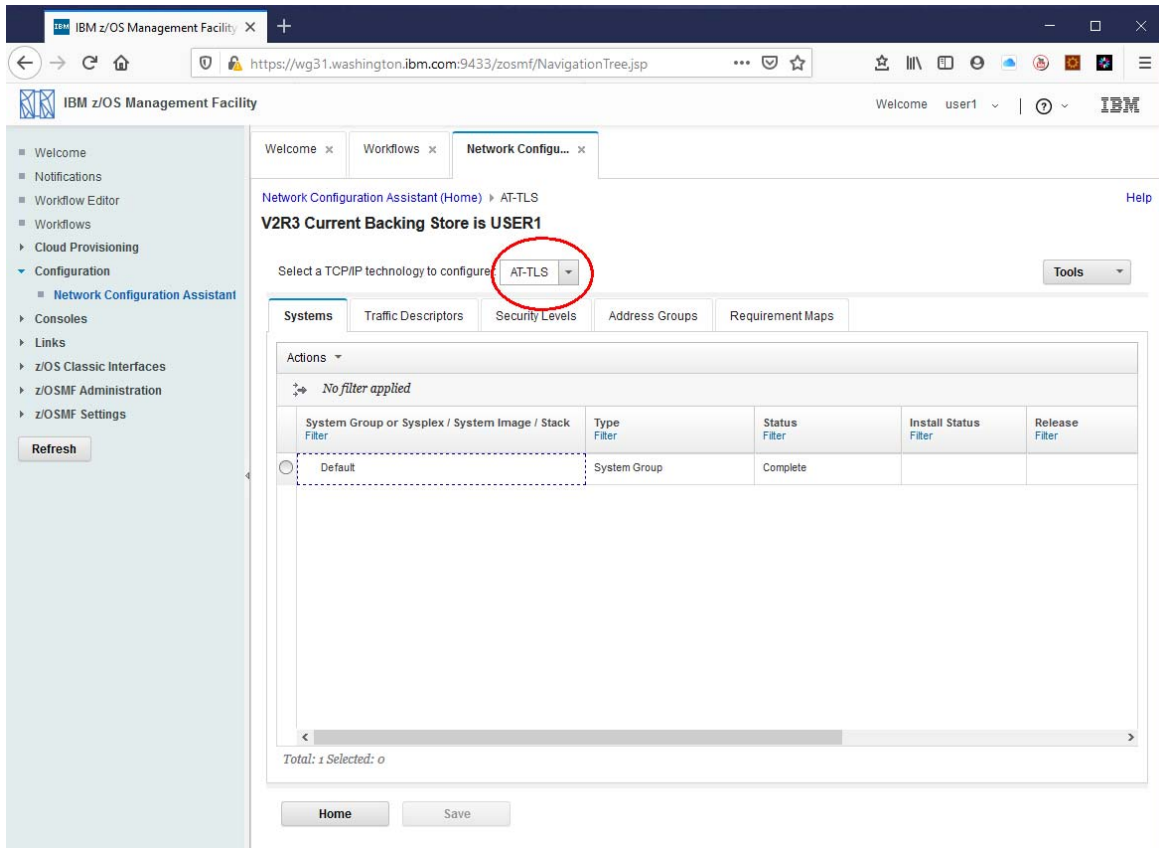
2. Enter *USER1* as the *z/OS USER ID* and *USER1*'s password and click the **LOG IN** button.

- \_\_\_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.

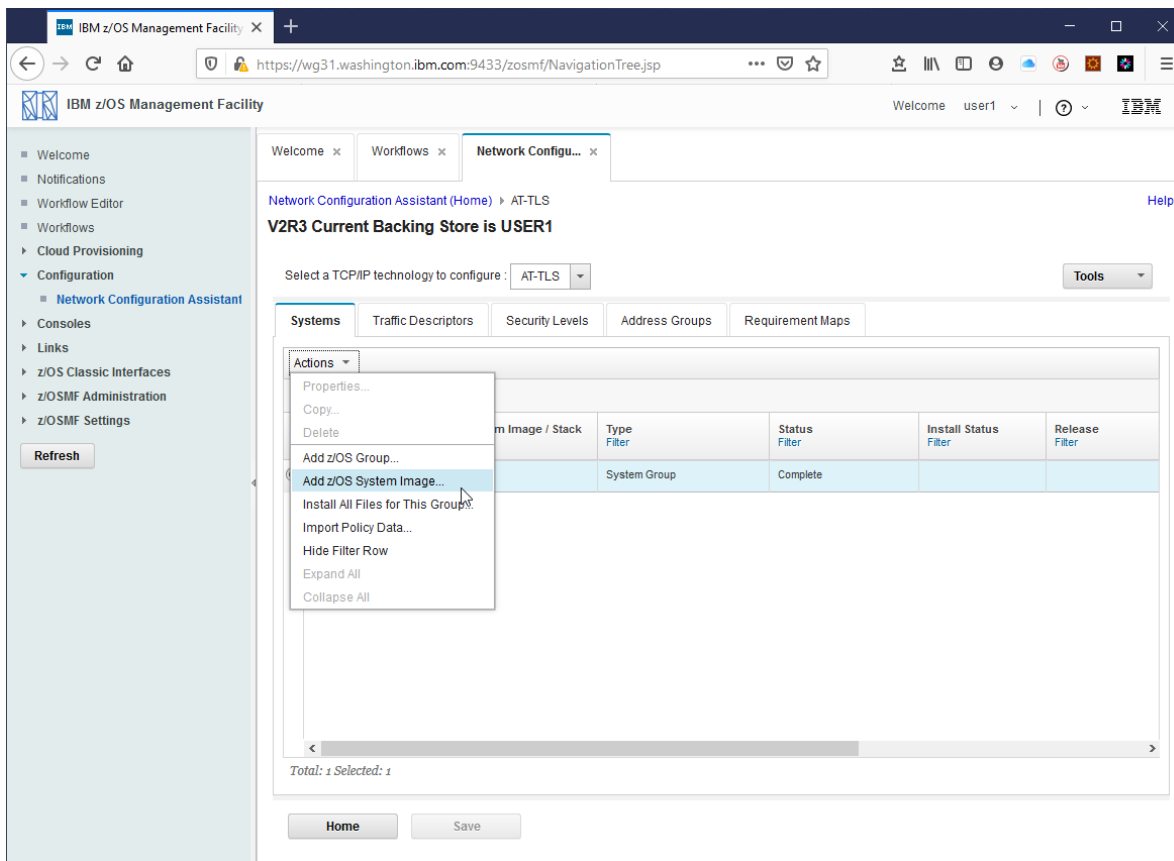


- \_\_\_4. Select the radio button beside *Create or transfer a new backing store* option and click the **Proceed** button.
- \_\_\_5. On the next screen select the radio button beside *Create a New Backing Store File* and enter **USER1** in the area beside *File Name*. Press the **OK** button and press 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.



- \_\_\_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...)*. Enter **Liberty.KeyRing** as the default AT-TLS key ring name. Click **OK** to continue.

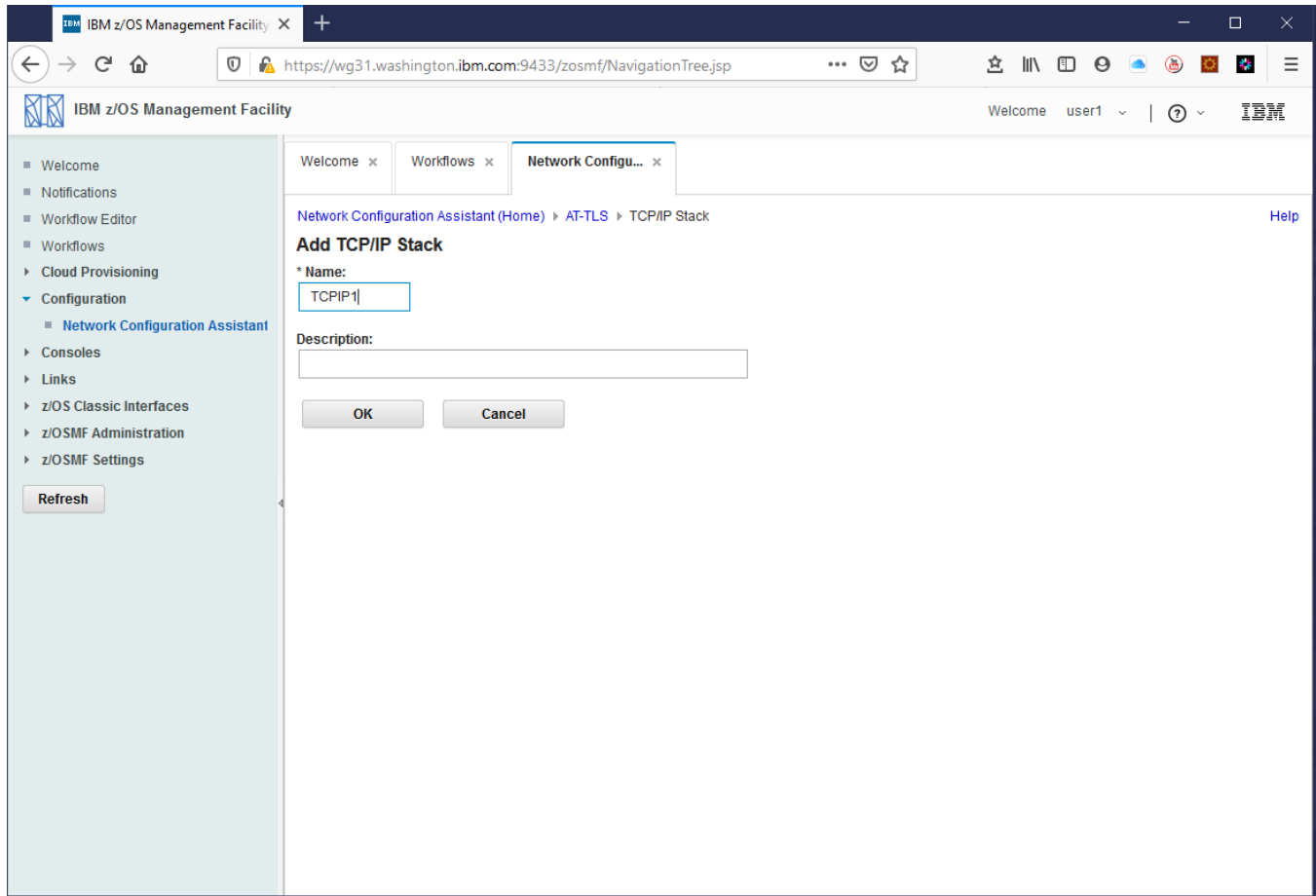
The screenshot shows the 'Add z/OS System Image' dialog in the IBM z/OS Management Facility. The 'Name' field is set to 'WG31'. The 'z/OS Release' is set to 'V2R3'. Under the 'Default AT-TLS key ring database' section, the 'Simple name (as in an SAF product or in PKCS #11 token format)' radio button is selected, and the 'Key ring' field is set to 'Liberty.KeyRing'. The 'OK' button is highlighted.

**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 by using the same key ring name. 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.



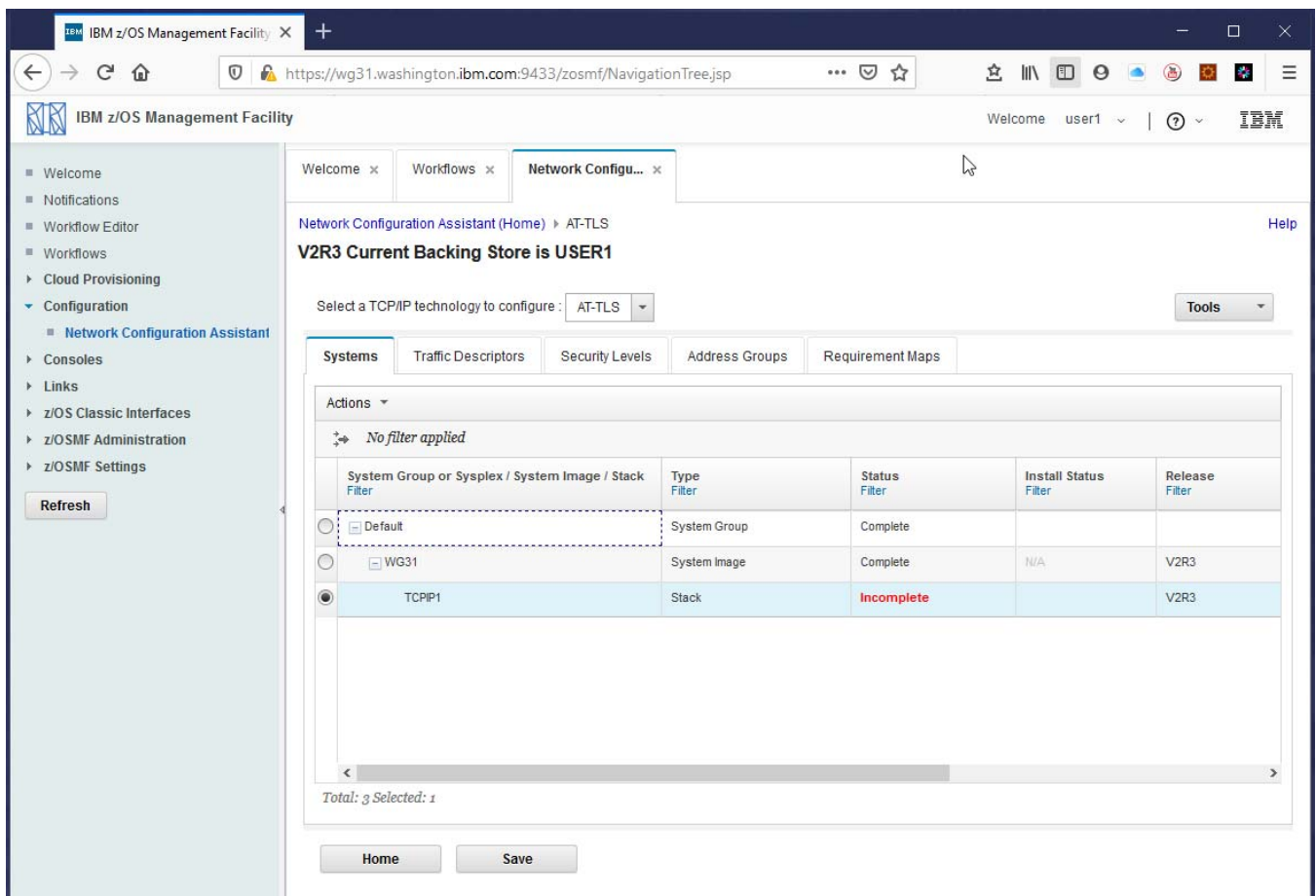
**Tech-Tip:** The value for the stack name was determined by the TCPIP Name display by entering the MVS command D TCPIP.

```
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.

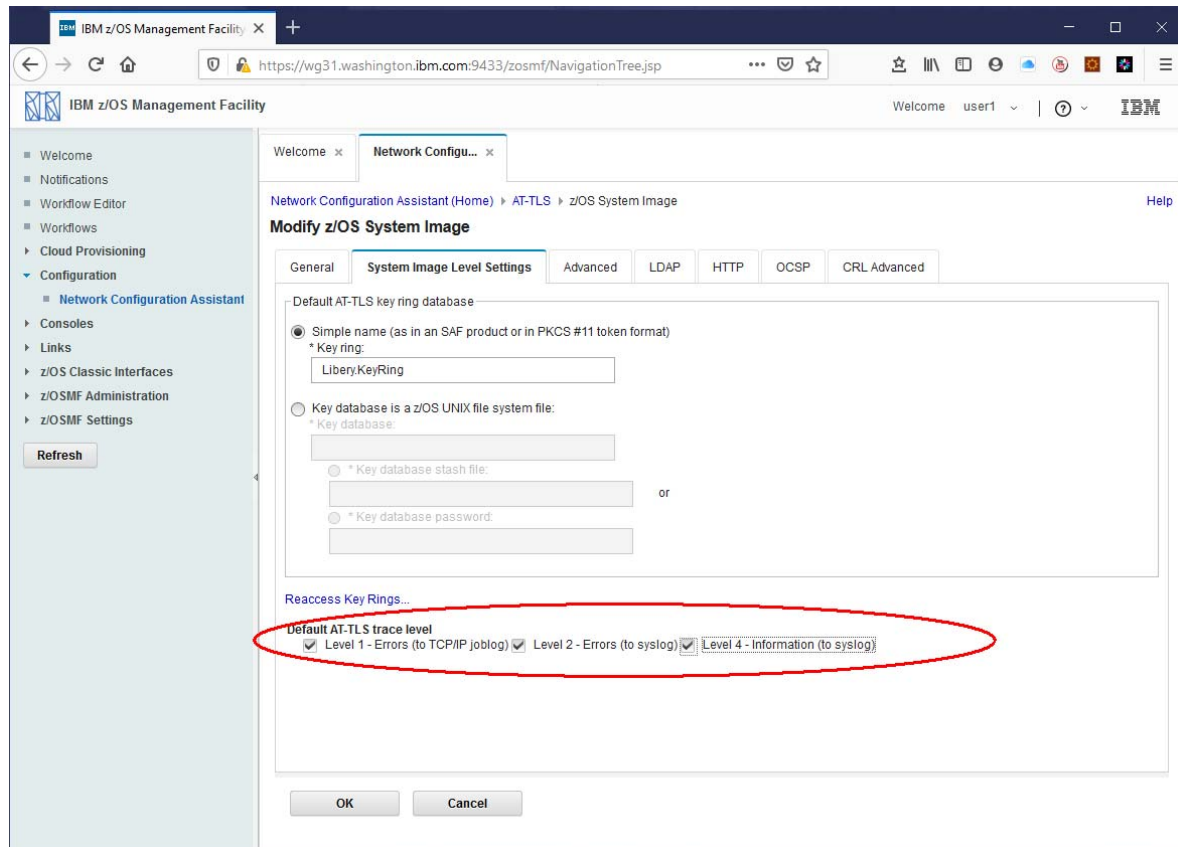


- \_\_\_12. This will display the window below:



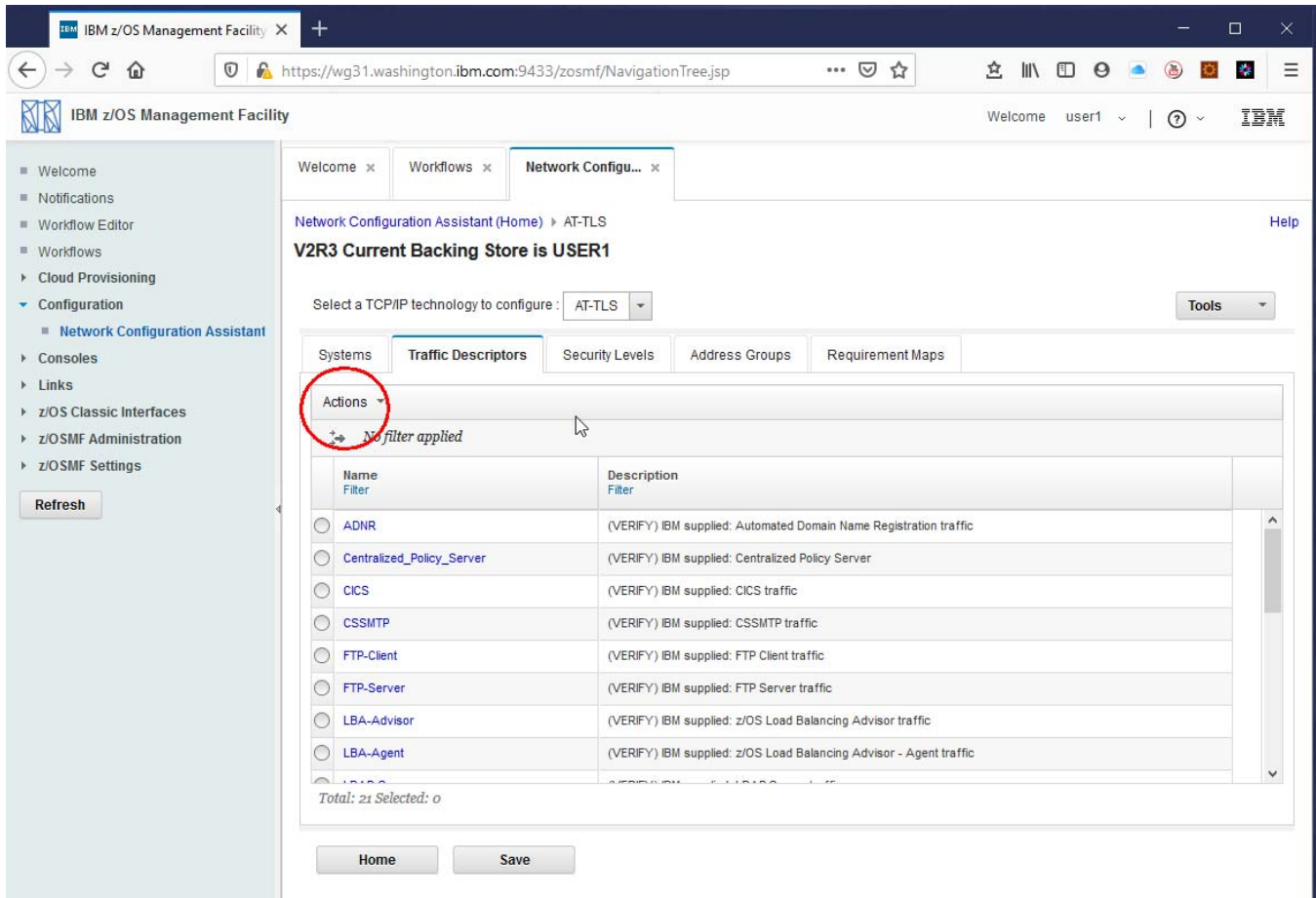
**Tech Tip:** The **Incomplete** warning will be addressed shortly.

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. 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.





\_\_\_14. Select the *Traffic Descriptors* tab and use the *Actions* pull-down to select *New*.



IBM z/OS Management Facility

Welcome user1

Network Configuration Assistant (Home) > AT-TLS

V2R3 Current Backing Store is USER1

Select a TCP/IP technology to configure: AT-TLS

Tools

Systems Traffic Descriptors Security Levels Address Groups Requirement Maps

Actions

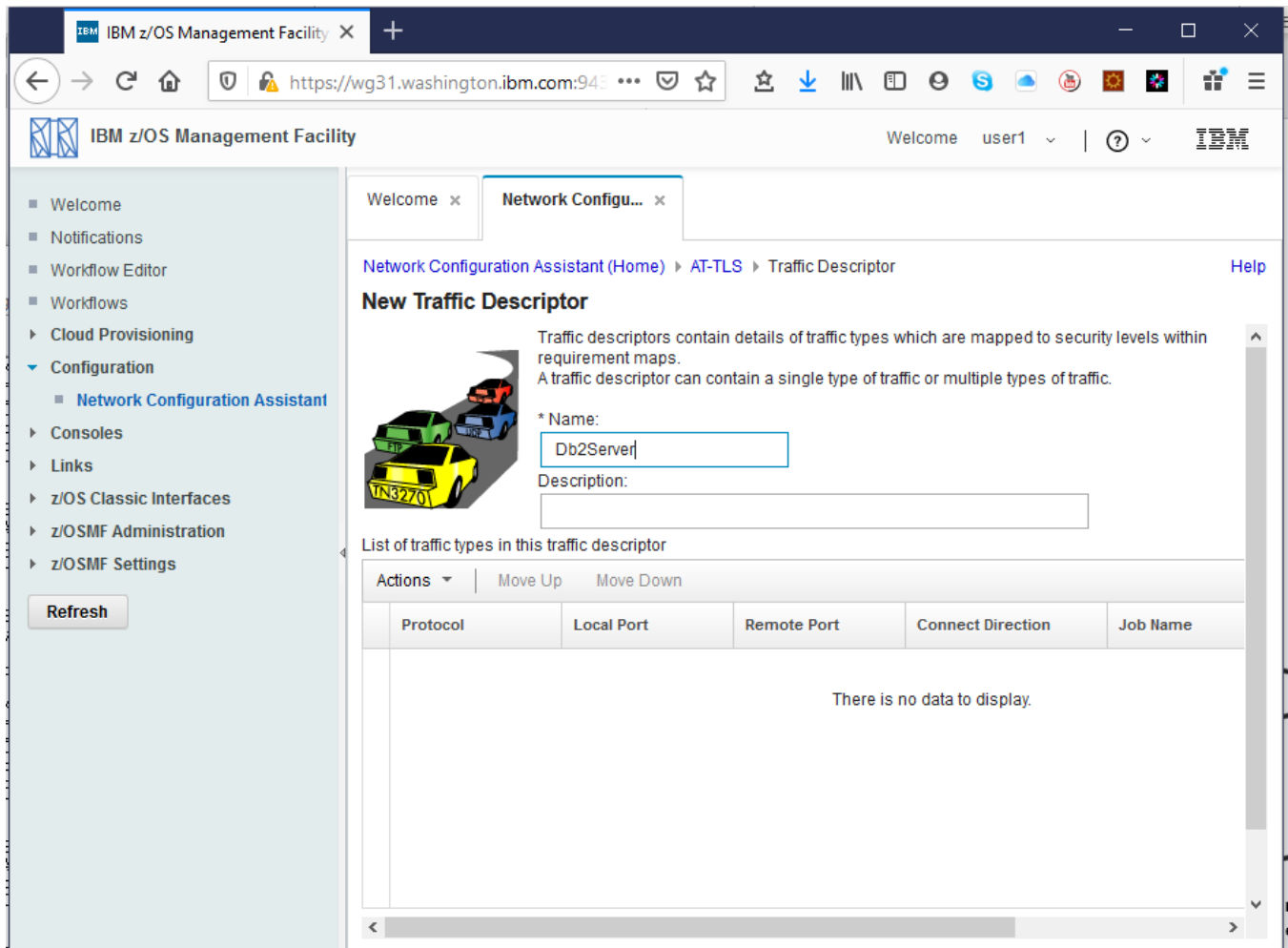
No filter applied

Name Filter	Description Filter
<input type="radio"/> ADNR	(VERIFY) IBM supplied: Automated Domain Name Registration traffic
<input type="radio"/> Centralized_Policy_Server	(VERIFY) IBM supplied: Centralized Policy Server
<input type="radio"/> CICS	(VERIFY) IBM supplied: CICS traffic
<input type="radio"/> CSSMTP	(VERIFY) IBM supplied: CSSMTP traffic
<input type="radio"/> FTP-Client	(VERIFY) IBM supplied: FTP Client traffic
<input type="radio"/> FTP-Server	(VERIFY) IBM supplied: FTP Server traffic
<input type="radio"/> LBA-Advisor	(VERIFY) IBM supplied: z/OS Load Balancing Advisor traffic
<input type="radio"/> LBA-Agent	(VERIFY) IBM supplied: z/OS Load Balancing Advisor - Agent traffic

Total: 21 Selected: 0

Home Save

15. On the *New Traffic Descriptor* window, enter **Db2Server** as the *Name*. 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 *Single ports* under *Local port* and enter **2445** as the port number. Select the radio button *All ports* under *Remote port*. Select the radio button beside *Inbound only* under *Indicate the TCP connection direction*. Enter **DSN2DIST** in the area under *Jobname* and finally select the radio button beside *Server* under *AT-TLS Handshake Role*. Next click on the *KeyRing* tab to continue.

The screenshot shows the IBM z/OS Management Facility Network Configuration Assistant (Home) window. The 'New Traffic Type - TCP' configuration screen is displayed. The 'Details' tab is active. The 'Local port' section has 'Single port' selected with the value '2,445'. The 'Remote port' section has 'All ports' selected. The 'Indicate the TCP connect direction' section has 'Inbound only' selected. The 'Jobname' field contains 'DSN2DIST'. The 'AT-TLS Handshake Role' section has 'Server' selected. The 'KeyRing' tab is visible but not active.

**Tech-Tip:** This traffic definition is triggered when a client attempts to connect to port 2445. Port 2445 was identified as the Db2 SECPORT in startup message.

```

DSNL004I -DSN2 DDF START COMPLETE
LOCATION DSN2LOC
LU      USIBMWZ.DSN2APPL
GENERICLU -NONE
DOMAIN  WG31.WASHINGTON.IBM.COM
TCPPOINT 2446
SECPOINT 2445
RESPORT 2447
IPNAME   -NONE
OPTIONS:
PKGREL = COMMIT

```

If all the defined conditions are met, AT-TLS will act as a surrogate for the server 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, e.g. Db2.KeyRing. Otherwise the default is to use the same key ring name defined for the z/OS System image, e.g. Liberty.KeyRing.

- \_\_\_17. On the *KeyRing* tab, select the radio button beside *Use a Simple name* and enter **Db2.KeyRing** as the key ring name. Click **OK** twice to continue.

IBM z/OS Management Facility

Network Configuration Assistant (Home) > AT-TLS > Traffic Descriptor > Traffic Type - TCP

**Modify Traffic Type - TCP**

Details KeyRing Advanced

Specify the key ring database to use for the traffic type specified on the Details tab.

☐ Use the key ring database defined for the z/OS system image.

☒ Use a Simple name (as in a SAF product or in PKCS #11 Token format):

\* Key ring:

Db2 KeyRing

☐ Use this z/OS UNIX file system key database:

\* Key database:

☐ \* Key database stash file:

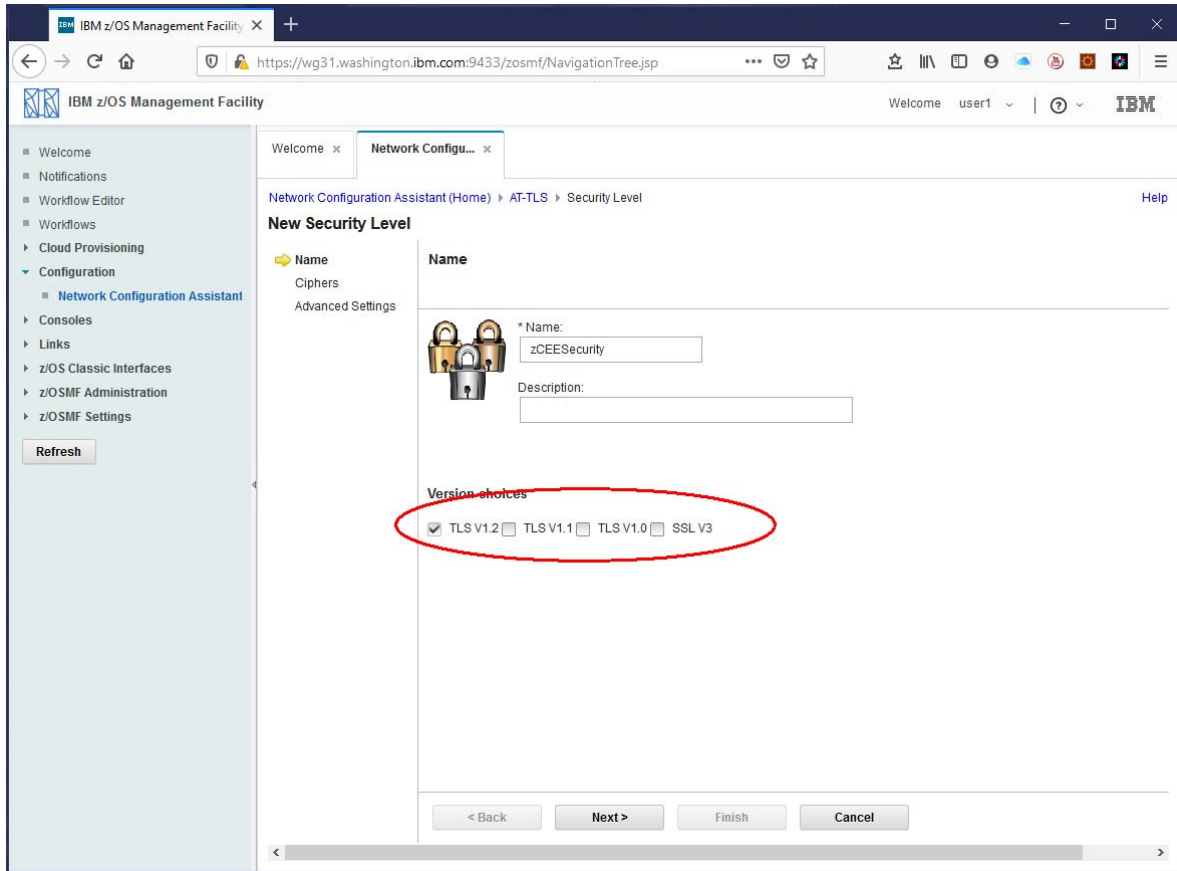
☐ \* Key database password:

Certificate Label:

☐ Specified server certificate labels to be used by server to accommodate clients with different types of public keys. The certificates are available beginning with V2R3.

Actions	Certificate Label
<input type="radio"/>	
<input type="radio"/>	
<input type="radio"/>	
<input type="radio"/>	

- \_\_\_18. Next, click the Security Levels tab and use the *Actions* pull-down button and 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.



**Tech Tip:** The key ring specified here belongs to identity DB2USER This is the identity under which the z/OS Connect server is running. This ring has these certificates connected.

```
>Db2.KeyRing<
Certificate Label Name      Cert Owner      USAGE      DEFAULT
-----
DB2 CA                     CERTAUTH        CERTAUTH    NO
zCEE CA                    CERTAUTH        CERTAUTH    NO
DB2USER                    ID (DB2USER)    PERSONAL     YES
```

The z/OS Connect out bound JSSE key ring has these certificates connected.

Ring:

```
>zCEE.KeyRing<
Certificate Label Name      Cert Owner      USAGE      DEFAULT
-----
zCEE CA                    CERTAUTH        CERTAUTH    NO
Liberty CA                 CERTAUTH        CERTAUTH    NO
zCEE Client Cert           ID (LIBSERV)    PERSONAL     YES
zCEE-CertAuth              CERTAUTH        CERTAUTH    NO
DB2 CA                     CERTAUTH        CERTAUTH    NO
```

\_\_\_19. Next, click the *Requirement Maps* tab. Use the *Actions* pull-down button to select the *New* option.

The screenshot shows the IBM z/OS Management Facility Network Configuration Assistant (Home) - AT-TLS page. The page has a left sidebar with navigation links: Welcome, Notifications, Workflow Editor, Workflows, Cloud Provisioning, Configuration (selected), Consoles, Links, z/OS Classic Interfaces, z/OSMF Administration, and z/OSMF Settings. The main content area shows the 'Requirement Maps' tab selected. The 'Actions' pull-down menu is open, showing 'No filter applied'. Below the menu is a table with two columns: Name and Description. The table contains one entry: 'AT-TLS\_Sample' with description 'IBM supplied: AT-TLS sample: CICS and TN3270'. The 'Total: 1 Selected: 1' status is shown at the bottom of the table. The 'Home' and 'Save' buttons are visible at the bottom of the page.

Name Filter	Description Filter
AT-TLS_Sample	IBM supplied: AT-TLS sample: CICS and TN3270

Total: 1 Selected: 1

20. On the *New Requirement Map* window, enter **Db2RequirementMap** as the *Name*. Use the pull-down arrows to select **Db2Server** as the *Traffic Descriptor* and **zCEESecurity** as the *Security Level* for this map. Click **OK** to continue.

IBM z/OS Management Facility

Welcome user1 | ? | IBM

Network Configuration Assistant (Home) > AT-TLS > Requirement Map

### New Requirement Map

A requirement map is an object that maps each IP traffic type (traffic descriptor) to a specific level of security (security level).

To add a new mapping to the requirement map:

1. Click the "Add Row" action or use an existing row
2. Click a table cell to select a traffic descriptor from the list
3. Click a table cell to select a security level from the list

\* Name:

Description:

Mappings table

Actions		Move Up	Move Down
	Traffic Descriptor	Security Level	
<input type="radio"/>	Db2Server	zCEESecurity	
<input type="radio"/>	Select a traffic descriptor	Select a security level	
<input type="radio"/>	Select a traffic descriptor	Select a security level	

Total: 3 Selected: 0

- \_\_\_21. Select the radio button beside *Db2RequirementMap* 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.

The screenshot shows the IBM z/OS Management Facility interface. The left sidebar contains a navigation menu with options like Welcome, Notifications, Workflow Editor, Workflows, Cloud Provisioning, Configuration (selected), Consoles, Links, z/OS Classic Interfaces, z/OSMF Administration, and z/OSMF Settings. The main content area is titled 'Network Configuration Assistant (Home) > AT-TLS > View Details'. It features a 'View Details' section with 'Close' and 'Printable page' buttons. Below this is the 'Application / Requirement Map: Db2RequirementMap' section, which includes a 'Requirement map summary' table and a 'Requirement Map traffic - Shown in Configured Order' table. The summary table shows 'Traffic Descriptor' as 'AT-TLS Security Level' and 'Db2Server' as 'zCEESecurity'. The traffic table lists 'Db2Server' using 'TCP' on '2445' port, with 'All Ports' as the remote port and 'Inbound' as the connect direction. The security level is 'zCEESecurity'. Below the traffic table is the 'Security Level Details' section, which specifies the security level as 'zCEESecurity', type as 'AT-TLS', encryption as 'System SSL V3 Defaults', and TLS version as '1.0'. At the bottom are 'Close' and 'Back to Top' buttons.

Requirement map summary					
Traffic Descriptor	AT-TLS Security Level				
Db2Server	zCEESecurity				

Requirement Map traffic - Shown in Configured Order						
Traffic Descriptor					AT-TLS Security Level	
Name	Protocol	Local Port	Remote Port	Connect Direction	Name	Ciphers
Db2Server	TCP	2445	All Ports	Inbound	zCEESecurity	---

**Security Level Details**

**Security Level: zCEESecurity**

Type: AT-TLS  
 Encryption: System SSL V3 Defaults  
 Use TLS Version 1.0:

- \_\_\_22. Click the **Save** button to save the configuration.



\_\_\_23. When the save has completed, click on the *Systems* tab to return to this window.

IBM z/OS Management Facility

Welcome user1

Network Configuration Assistant (Home) > AT-TLS

V2R3 Current Backing Store is USER1

Select a TCP/IP technology to configure: AT-TLS

Tools

Systems Traffic Descriptors Security Levels Address Groups Requirement Maps

Actions

No filter applied

System Group or Sysplex / System Image / Stack	Type	Status	Install Status	Release
Default	System Group	Complete		
WG31	System Image	Complete	N/A	V2R3
TCP/IP1	Stack	Incomplete		V2R3

Total: 3 Selected: 1

Home Save

24. Select the radio button beside *TCPIPI* and use the *Actions* pull-down to select *Rules*. This is where these definitions will be tied together. Use the *Actions* pull-down again and select *New* to create a new connectivity rule. Enter ***Db2ServerRule*** for the *Connectivity rule name* and press **Next** to continue.

The screenshot shows the IBM z/OS Management Facility interface. The left sidebar contains a navigation menu with options like Welcome, Notifications, Workflow Editor, Workflows, Cloud Provisioning, Configuration (selected), Consoles, Links, z/OS Classic Interfaces, z/OSMF Administration, and z/OSMF Settings. The main content area is titled 'Network Configuration Assistant (Home) > AT-TLS > TCP/IP Stack > Connectivity Rule'. Below this, the 'New Connectivity Rule' screen is displayed. It has a left sidebar with 'Data Endpoints' (selected), 'Requirement Map', and 'Advanced Settings'. The main area is titled 'Data Endpoints' and contains the following fields:

- \* Connectivity rule name:
- Select the address groups of the host endpoints of the traffic you want to protect.
- Local data endpoint:
  - ☒ Address group:
  - ☐ \* IPv4 or IPv6 address, subnet, or range:
  - Examples: xxxx, xxxxyy, xxxx-yy.yy, xx, xx/yyy, xx-y:y
- Remote data endpoint:
  - ☒ Address group:
  - ☐ \* IPv4 or IPv6 address, subnet, or range:
  - Examples: xxxx, xxx, xx, xx/yyy, xx-y:y

At the bottom, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

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 *Db2RequirementMap*. This should automatically populate the mapping table with *Db2Server* as the traffic descriptor and *zCEESecurity* as the security level. Press **Next** and then **Finish** to continue.

IBM z/OS Management Facility

Welcome user1

Network Configuration Assistant (Home) > AT-TLS > TCP/IP Stack > Connectivity Rule

### New Connectivity Rule

- ✓ Data Endpoints
- ➔ Requirement Map
- Advanced Settings

#### Requirement Map

Requirement maps are reusable objects that combine your traffic definitions (traffic descriptors) with your security definitions (security levels).

☐ Create a new requirement map  
☒ Select an existing requirement map

Db2RequirementMap

#### Db2RequirementMap properties

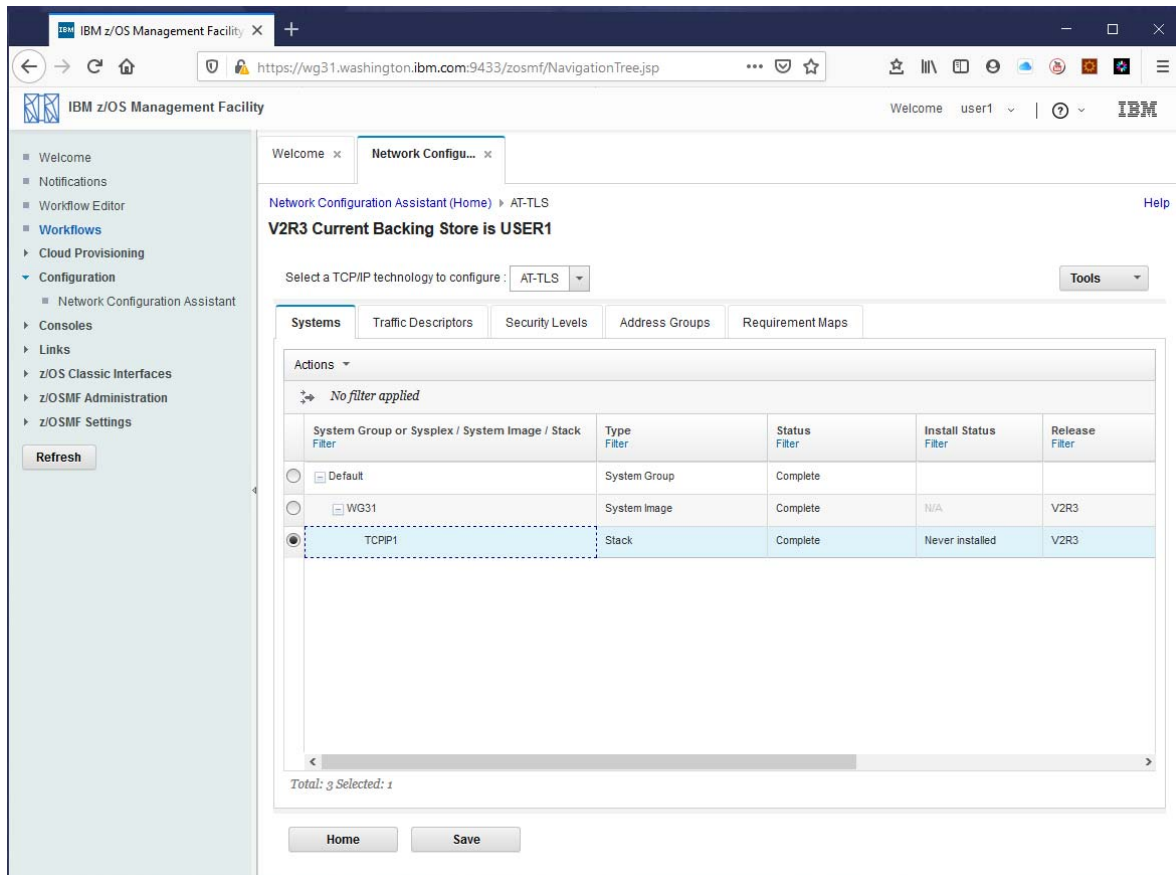
\* Name: Db2RequirementMap

Description:

Traffic Descriptor	Security Level
Db2Server	zCEESecurity

< Back   Next >   Finish   Cancel

\_\_\_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*.

IBM z/OS Management Facility

Welcome user1

Network Configuration Assistant (Home) > AT-TLS > Configuration Files

**List of Configuration Files for All System Images In Group Default**

List of Configuration Files for All System Images In Group Default

System Image	Configuration Type	Status	Last Install	Configured File Name	Configured Host Name	Configured Installation Method
WG31	TCP/IP1 - AT-TLS Policy	Never installed	Never	/etc/cf/gasst /v2r3/WG31 /TCP/IP1.tlsPol		Save to disk

Total: 1 Selected: 1

Close

\_\_\_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 it is not active yet.

## 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")/-I 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 the bottom of the file and you will see something like what is shown below:

```
WG31
File Edit Settings View Communication Actions Window Help
File Edit Edit Settings Menu Utilities Compilers Test Help

VIEW /SYSTEM/var/syslogd/syslogall.log Columns 00063 00134
Command ==> Scroll ==> 4
003388 YFT18I Using catalog '/usr/lib/nls/msg/C/ftpdmsg.cat' for FTP messages.
003389 Y2697I IBM FTP CS V2R3 19:44:07 on 03/23/20
003390 Y2640I Using dd:SYSFTPD=SYS1.TCPPARMS(FTPDATA) for local site configurat
003391 YFT47I dd:SYSFTPD=SYS1.TCPPARMS(FTPDATA) file, line 10: Ignoring keyword
003392 YFT47I dd:SYSFTPD=SYS1.TCPPARMS(FTPDATA) file, line 11: Ignoring keyword
003393 YFT47I dd:SYSFTPD=SYS1.TCPPARMS(FTPDATA) file, line 13: Ignoring keyword
003394 YFT47I dd:SYSFTPD=SYS1.TCPPARMS(FTPDATA) file, line 49: Ignoring keyword
003395 YFT47I dd:SYSFTPD=SYS1.TCPPARMS(FTPDATA) file, line 54: Ignoring keyword
003396 YFT21I Using catalog '/usr/lib/nls/msg/C/ftpdprly.cat' for FTP replies.
003397 YFT26I Using 7-bit conversion derived from 'ISO8859-1' and 'IBM-1047' fo
003398 YFT32I Using the same translate tables for the control and data connecti
003399 YFT09I system information for WG31: z/OS version 2 release 3 (3906)
003400 pFixLevel: Fix level: NONEFND Data: EZB0ECPR
003401 pFixLevel: Fix level: HIP6230 Data: EZAFTPDA EZAFTPDI EZAFTPF4 EZAFTPGA
003402 pFixLevel: Fix level: " Data: EZAFTPG1 EZAFTPXC EZAFTPB8 EZAFTPDF
003403 pFixLevel: Fix level: " Data: EZAFTPDH EZAFTPDM EZAFTPEA EZAFTPED
003404 pFixLevel: Fix level: " Data: EZAFTPEJ EZAFTPER EZAFTPET EZAFTPGU
003405 pFixLevel: Fix level: " Data: EZAFTPGV EZAFTPNX EZAFTPSD EZAITUTI
003406 pFixLevel: Fix level: UI53145 Data: EZAFTPNY
003407 pFixLevel: Fix level: UI56159 Data: EZAFTPEP
003408 pFixLevel: Fix level: UI57631 Data: EZAFTPF5
003409 pFixLevel: Fix level: 24/ 24 Data: OBJECTS PROCESSED. AV-BUFR: 0005087
003410 Y2700I Using port FTP control (21)
003411 Y2701I Inactivity time is 12000
003412 Y2702I Server-FTP: Initialization completed at 19:44:07 on 03/23/20.
003413 YFT41I Server-FTP: process id 83886182, server job name FTPSERVE
003414 nning on 0.0.0.0 port 22.
***** Bottom of Data *****
MA C 04/015
Connected to remote server/host wg31 using lu/pool TCP00109 and port 23
```

- \_\_\_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 sting *EZZ8431I PAGENT STARTING*. 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 TTLSRule* 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 Db2ServerRule~1'
003517 Processing: 'TTLSGroupAction gAct1~Db2Server'
003518 Processing: 'TTLSEnvironmentAction eAct1~Db2Server'
003519 Processing: 'TTLSConnectionAction cAct1~Db2Server'
003520 Processing: 'TTLSConnectionAdvancedParms cAdv1~Db2Server'
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~ Db2Server'
003527 Processing TTLS Connection action 'cAct1~Db2Server'
003528 Processing TTLS Environment action 'eAct1~Db2Server'
003529 ocessing TTLS rule 'Db2Server~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
```



**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. On this image this has been configured this way so the AT-TLS is disabled by default and must be explicitly enabled. This is done by an MVS VARY command.

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

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

Issue this command and you should see these messages in the console.

```
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. Edit the *server.xml* configuration file for the *myServer* server, e.g. */var/zosconnect/servers/myServer/server.xml* and change the include for file *db2passTicket.xml* to an include of file *db2TLS.xml*, see below.

**<include location="\$\${server.config.dir}/includes/db2TLS.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/imsDatabase.xml"/>
<include location="$${server.config.dir}/includes/db2TLS.xml"/>
<include location="$${server.config.dir}/includes/shared.xml"/>
```

The contents of *db2TLS.xml* are shown below.

```
<zoscconnect_zosConnectServiceRestClientConnection id="Db2Conn"
  sslCertsRef="OutboundSSLSettings"
  host="wg31.washington.ibm.com"
  port="2445"
  basicAuthRef="dsn2Auth" />

<zoscconnect_zosConnectServiceRestClientBasicAuth id="dsn2Auth"
  applName="DSN2APPL" />
```

\_\_\_10. Refresh the z/OS Connect server's profile with MVS command **F BAQSTRT,REFRESH,CONFIG**

## *Test the TLS connection from the zCEE Server to a Db2 subsystem*

\_\_\_1. Open a DOS command prompt and go to directory *c:/z/admin*.

\_\_\_2. Enter the cURL commands below:

```
curl -X get --cacert certauth.pem --cert fred.p12:secret --cert-type P12  
https://wg31.washington.ibm.com:9443/db2/employee/000010
```

```
curl -X get --cacert certauth.pem --cert user1.p12:secret --cert-type P12  
https://wg31.washington.ibm.com:9443/db2/employee/000010
```

You should see the same results as before. A request from FRED does not work because when the pass ticket for FRED is sent to Db2 this user does not have sufficient authority. When the pass ticket for USER1 is sent to DB2 this user has sufficient authority to access Db2.

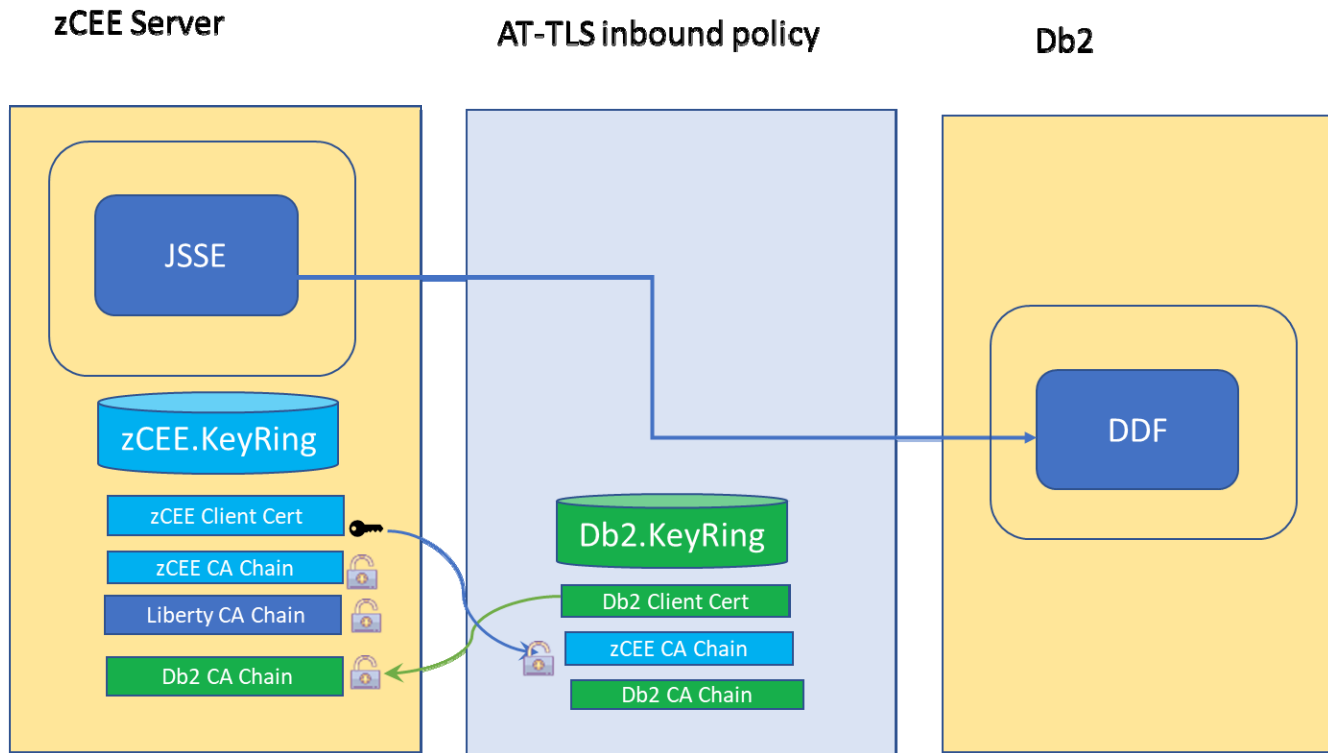
\_\_\_3. View the *syslogall.log* file again and you should find two sets of the messages shown below. Each set of messages showing the AT-TLS policy has been triggered and the message sent from the z/OS Connect server has been protected with TLS. The handshakes were successful even though the request for FRED failed with Db2 security check.

```
EZD1281I TTLS Map CONNID: 000029DB LOCAL: 192.168.17.220..2445 REMOTE:  
192.168.17.220..9844 JOBNAME: DSN2DIST USERID: DB2USER TYPE: InBound STATUS:  
Enabled RULE: Db2ServerRule~1 ACTIONS: gAct1~Db2Server eAct1~Db2Server cAct1~Db2Server  
EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000001 CONNID: 000029DB RC: 0 Initial  
Handshake 0000005011527E10 0000005011522750 TLSV1.2 F0F0F3F5
```

....

```
EZD1281I TTLS Map CONNID: 00002A69 LOCAL: 192.168.17.220..2445 REMOTE:  
192.168.17.220..9965 JOBNAME: DSN2DIST USERID: DB2USER TYPE: InBound STATUS:  
Enabled RULE: Db2ServerRule~1 ACTIONS: gAct1~Db2Server eAct1~Db2Server cAct1~Db2Server  
EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000003 CONNID: 00002A69 RC: 0 Initial  
Handshake 0000005011527E10 0000005011522750 TLSV1.2 F0F0F3F5
```

The diagram below shows the flow of the request from the z/OS Connect server to Db2.



\_\_\_4. Here is an example of a handshake failure. The request failed with a return code of 202.

The key ring cannot be opened because the user does not have permission. Check the following items:

- Look at message ESD1281 to verify the user ID being used for this connection and the TTLSEnvironmentAction statement that is mapped to this connection. If you are configuring by using the IBM Configuration Assistant for z/OS® Communications Server, you can specify the key ring on either the AT-TLS: Image Level Settings panel or on each Traffic Descriptor.
- Ensure that the correct key ring is specified

- \_\_\_5. The information in the AT-TLS Knowledge Center for describing AT-TLS return codes has the information displayed below for a 202 return code (see URL [https://www.ibm.com/support/knowledgecenter/SSLTBW\\_2.4.0/com.ibm.zos.v2r4.hald001/comtls.htm\\_](https://www.ibm.com/support/knowledgecenter/SSLTBW_2.4.0/com.ibm.zos.v2r4.hald001/comtls.htm_)

```
EZD1281I TTLS Map CONNID: 0000247F LOCAL: 192.168.17.220..2445 REMOTE :
192.168.17.220..8736 JOBNAME: DSN2DIST USERID: DB2USER TYPE: InBound STATUS:
Enabled RULE: Db2ServerRule~1 ACTIONS: gAct1~Db2Server eAct1~Db2Server cAct1~Db2Server
```

```
EZD1286I TTLS Error GRPID: 00000001 ENVID: 00000001 CONNID: 00000000 LOCAL:
**N/A** REMOTE: **N/A** JOBNAME: **N/A** USERID: DB2USER RULE: **N/A**
RC: 202 Environment Master Init 0000000000000000
```

```
EZD1286I TTLS Error GRPID: 00000001 ENVID: 00000001 CONNID: 00000000 LOCAL:
**N/A** REMOTE: **N/A** JOBNAME: **N/A** USERID: DB2USER RULE: **N/A**
RC: 202 Environment Link 0000000000000000 00000002
```

```
EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000001 CONNID: 0000247F RC: 5006 Initial
Handshake 0000000000000000 0000000000000000 0000000000000000 00000000
```

```
EZD1286I TTLS Error GRPID: 00000001 ENVID: 00000001 CONNID: 0000247F LOCAL:
192.168.17.220..2445 REMOTE: 192.168.17.220..8736 JOBNAME: DSN2DIST USERID:
DB2USER RULE: Db2ServerRule~1 RC: 5006 Initial Handshake 0000000000000000
0000000000000000 0000000000000000 00000000
```

This simply means the user does not have the required access to either of the RACF FACILITY resources IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTING

## Optional

Disable the use of RACF Passtickets and retest using only TLS and see what difference this makes.

## Summary

In this step you have created an AT-TLS inbound policy which protects Db2's port 2445. This policy has a server surrogate for Db2 for TLS handshakes.

## Appendix – AT-TLS Policy Agent Configuration File

```
##
## AT-TLS Policy Agent Configuration file for:
##   Image: WG31
##   Stack: TCPIP1
##
## Created by the IBM Configuration Assistant for z/OS Communications Server
## Version 2 Release 3
## Backing Store = USER1
## Install History:
## 2020-06-12 13:15:40 : Save To Disk
##
## End of Network Configuration Assistant information
TTLSRule                                Db2ServerRule~1
{
  LocalAddrSetRef                        addr1
  RemoteAddrSetRef                       addr1
  LocalPortRangeRef                      portR1
  RemotePortRangeRef                     portR2
  Jobname                                DSN2DIST
  Direction                              Inbound
  Priority                                255
  TTLSGroupActionRef                     gAct1~Db2Server
  TTLSEnvironmentActionRef                eAct1~Db2Server
  TTLSConnectionActionRef                 cAct1~Db2Server
}
TTLSGroupAction                          gAct1~Db2Server
{
  TTLSEnabled                            On
  Trace                                  7
}
TTLSEnvironmentAction                    eAct1~Db2Server
{
  HandshakeRole                           Server
  EnvironmentUserInstance                  0
  TTLSKeyringParmsRef                      keyR1
}
TTLSConnectionAction                     cAct1~Db2Server
{
  HandshakeRole                           Server
  TTLSConnectionAdvancedParmsRef           cAdv1~Db2Server
  CtraceClearText                          Off
  Trace                                    7
}
TTLSConnectionAdvancedParms              cAdv1~Db2Server
{
  SSLv3                                    Off
  TLSv1                                    Off
  TLSv1.1                                  Off
  SecondaryMap                             Off
  TLSv1.2                                  On
}
TTLSKeyringParms                         keyR1
{
  Keyring                                  Db2.KeyRing
}
```

```
IpAddrSet          addr1
{
  Prefix           0.0.0.0/0
}
PortRange          portR1
{
  Port             2445
}
PortRange          portR2
{
  Port             1024-65535
}
```