IBM z/OS Connect

Customization - Security and Db2

IBM Z
Wildfire Team –
Washington System Center

Lab Version Date: February 23, 2023

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.

Mitch Johnson (mitchj@us.ibm.com)

General Exercise Information and Guidelines

- ✓ This exercise requires the completion of the *IBM z/OS Connect Customization - Basic Configuration (1 of 2)* and *IBM z/OS Connect Customization - Basic Security (2 of 2)* 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.

- ✓ For information regarding the use of the Personal Communication 3270 emulator, see the *Personal Communications Tips* PDF in the exercise folder.

Mitch Johnson (mitchj@us.ibm.com)

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.

Mitch Johnson (mitchj@us.ibm.com)

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. 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="${shared.config.dir}/db2.xml"/>

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

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

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

___2. 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.

___3. Open a DOS command prompt and go to directory *c:/z/admin*.

Mitch Johnson (mitchj@us.ibm.com)

___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
{"StatusDescription":"Execution Successful","ResultSet
Output":[{"firstName":"CHRISTINE","lastName":"HAAS","middleInitial":"I","phoneNumber":
"3978","department":"A00","job":"PRES
","employeeNumber":"000010"}],"StatusCode":200}
```

___5. 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).

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

 <zosconnect_zosConnectServiceRestClientBasicAuth id="dsn2Auth"
       userName="USER1"
       password="USER1"/>
```

USER1 has READ access to DSNR resource DSN2.REST.

Mitch Johnson (mitchj@us.ibm.com)

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.

Mitch Johnson (mitchj@us.ibm.com)

___2. Edit the server's server.xml file and change the include for *db2.xml* to an include for ***db2passTicket.xml***.

<include location="${shared.config.dir}/db2passTicket.xml"/>

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

The contents of db2passTicket are shown below.

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

 <zosconnect_zosConnectServiceRestClientBasicAuth id="dsn2Auth"
       applName="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.

Mitch Johnson (mitchj@us.ibm.com)

___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.

Mitch Johnson (mitchj@us.ibm.com)

Configuring TSL 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 **ZCEETSLC** (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.

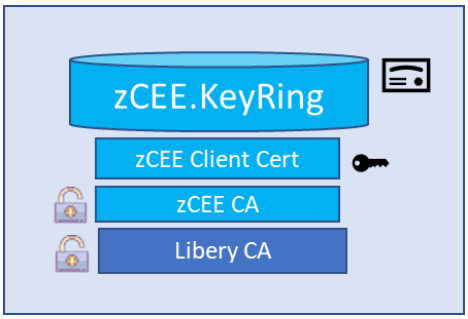
```
/* Create personal certificate for zCEE outbound client request */
racdcert id(libserv) gencert subjectsdn(cn('zCEE Client Cert') +
ou('ATS') o('IBM')) withlabel('zCEE Client Cert') signwith(certauth +
label('zCEE CA')) notafter(date(2022/12/31))

 /* 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 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)
```


Mitch Johnson (mitchj@us.ibm.com)

These commands

- Define a personal certificate for the z/OS Connect server for use during outbound handshakes.
- Define a key ring to be used for outbound handshakes.
- Connect the z/OS Connect server personal certificate to this key ring.
- Connect the certificate authority (CA) public certificate used to sign the z/OS Connect 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 z/OS Connect 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 z/OS Connect 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="/${shared.config.dir}/keyingOutboundMutual.xml"/>

Mitch Johnson (mitchj@us.ibm.com)

___4. Next browse member **DB2TLS**. The job contains the RACF commands below. Submit the job for execution.

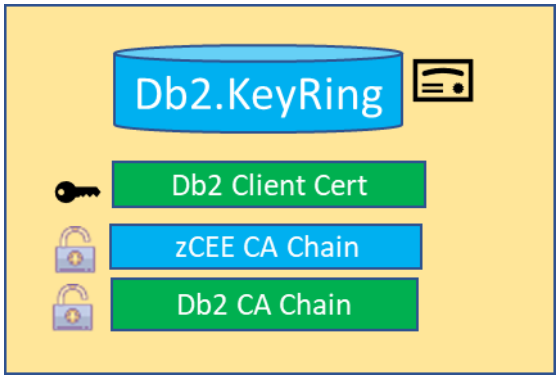
```
/* Create a CA certficate for DB2 */
racdcert certauth gencert subjectsdn(cn('Db2 CA') ou('ATS') +
ou('ATS') o('IBM')) withlabel('DB2 CA') keyusage(certsign) +
notafter(date(2022/12/31))

 /* Create a server certficate 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 certificiate */
racdcert id(db2user) connect(ring(Db2.KeyRing) +
          label('DB2USER') default

setropts raclist(digtring,digtnmap) refresh
```

These commands

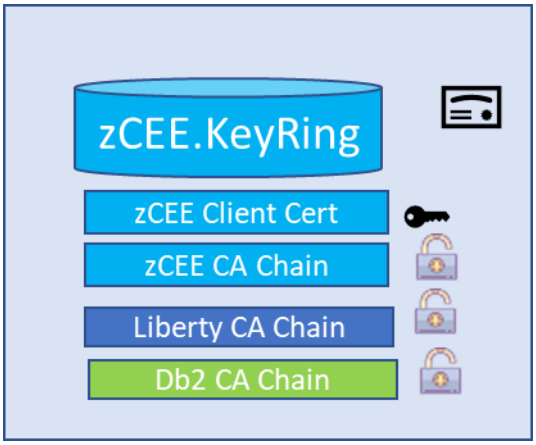
- Define a certificate authority certificate used to sign Db2 certificates used during TLS handshakes.
- Define a personal certificate for the Db2 server for use during TLS handshakes.
- Define a key ring to be used for TLS handshakes.
- Connect the Db2 server personal certificate to this key ring.
-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

Mitch Johnson (mitchj@us.ibm.com)

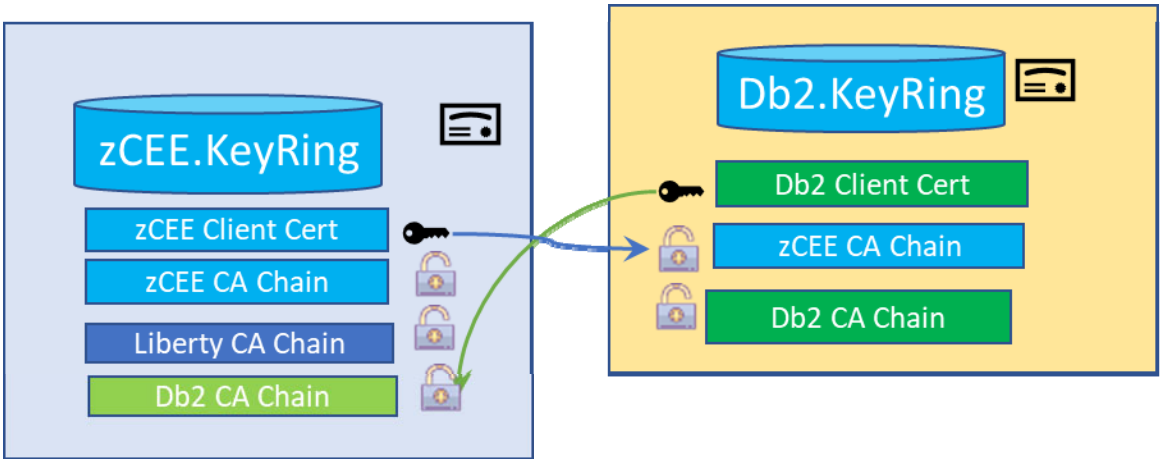
Below is a visual representation of the key ring just created:



And the update made to the z/OS Connect server's outbound keyring:



And the handshakes will flow as shown below:



Mitch Johnson (mitchj@us.ibm.com)

___5. 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="${shared.config.dir}/keyringOutboundMutual.xml"/>

```
<include location="${shared.config.dir}/safSecurity.xml"/>
<include location="${shared.config.dir}/ipicIDProp.xml"/>
<include location="${shared.config.dir}/keyringOutboundMutual.xml"/>
<include location="${shared.config.dir}/groupAccess.xml"/>
<include location="${shared.config.dir}/apiRequesterHTTPS.xml"/>
<include location="${shared.config.dir}/imsDatabase.xml"/>
<include location="${shared.config.dir}/db2passTicket.xml"/>
<include location="${shared.config.dir}/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" />
```


Mitch Johnson (mitchj@us.ibm.com)

___6. Enter MVS commands ***P BAQSRT and S BASSTRT*** to refresh the z/OS Connect server's runtime configuration.

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

F BAQSTRT,REFRESH,KEYSTORE

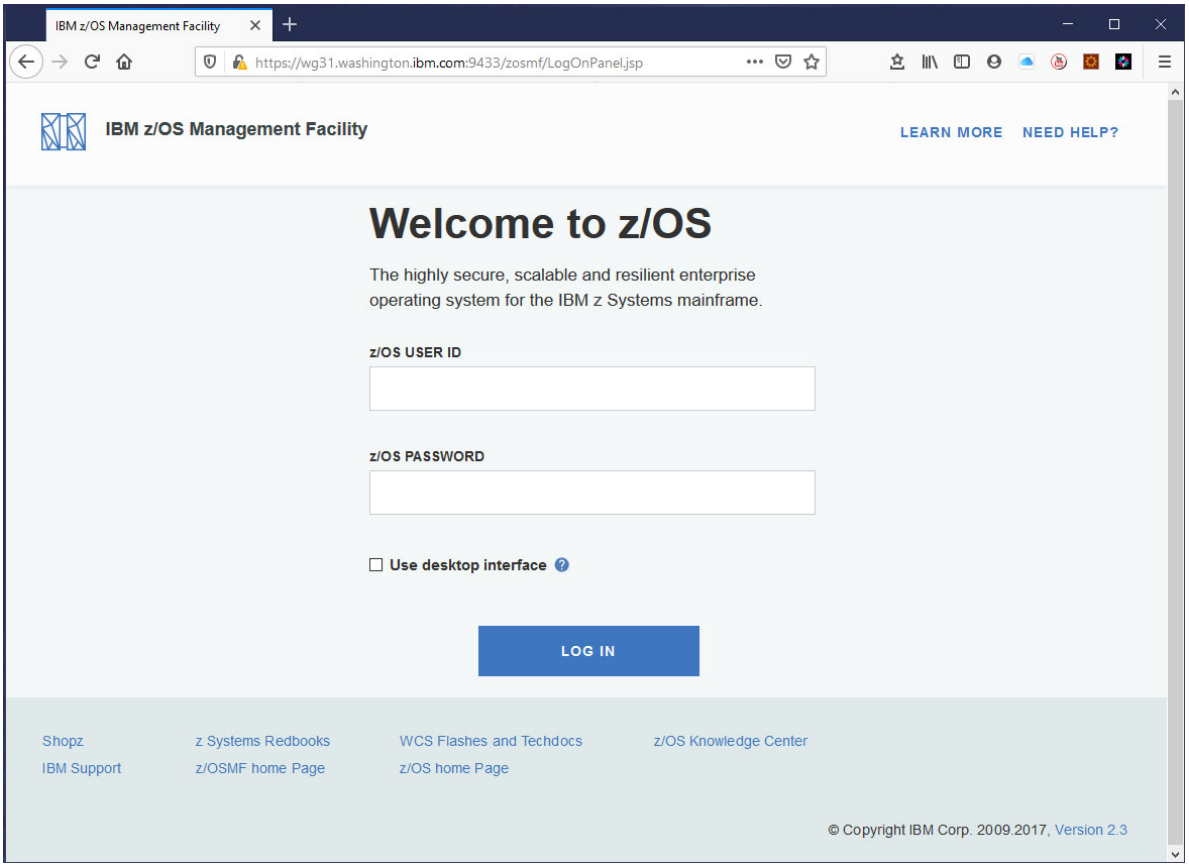
This would have dynamically made the information about CICS CA certificate available in the z/OS Connect runtime.

Mitch Johnson (mitchj@us.ibm.com)

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.

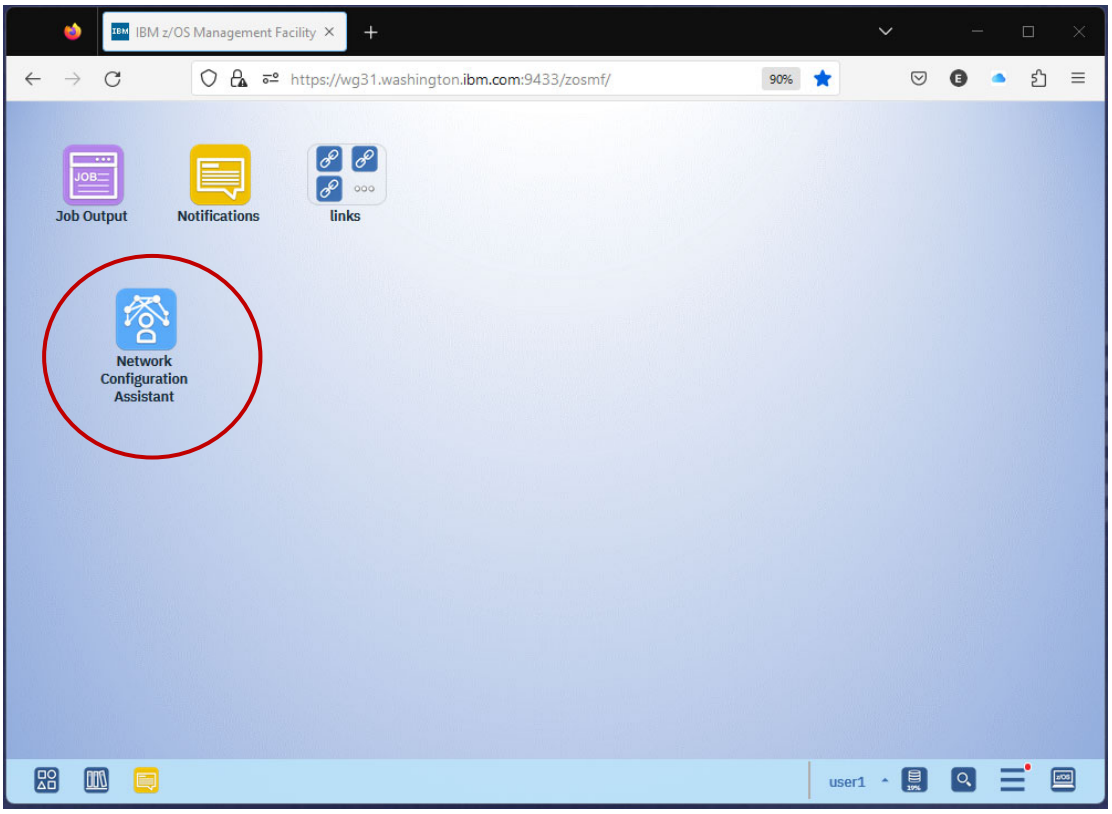


Note that some of the AT-TLS configurations steps described here may have been performed in another exercise.

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

Mitch Johnson (mitchj@us.ibm.com)

___3. On the initial zOSMF panel, select the *Network Configuration Assistant* by double clicking on the icon labeled *Network Configuration Assistant*.



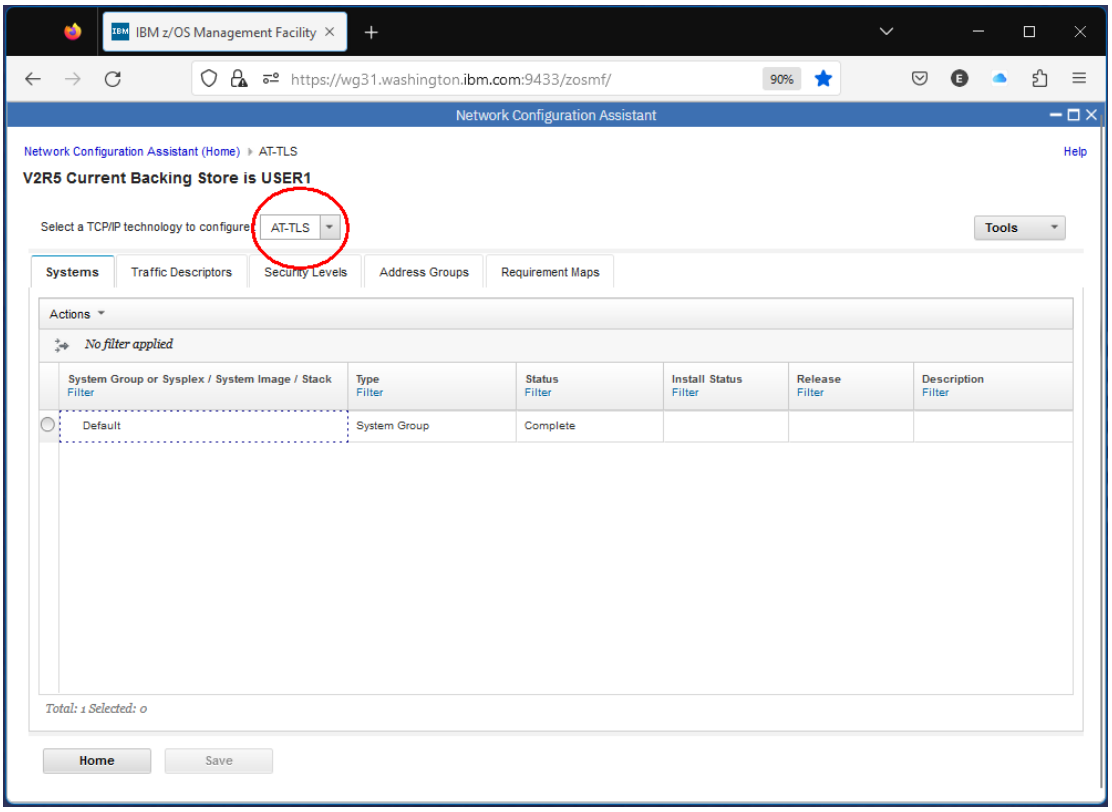
Tech-Tip: Subsequent screen shot show the *Network Configuration Assistance* expanded to fill the browser screen.

___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.

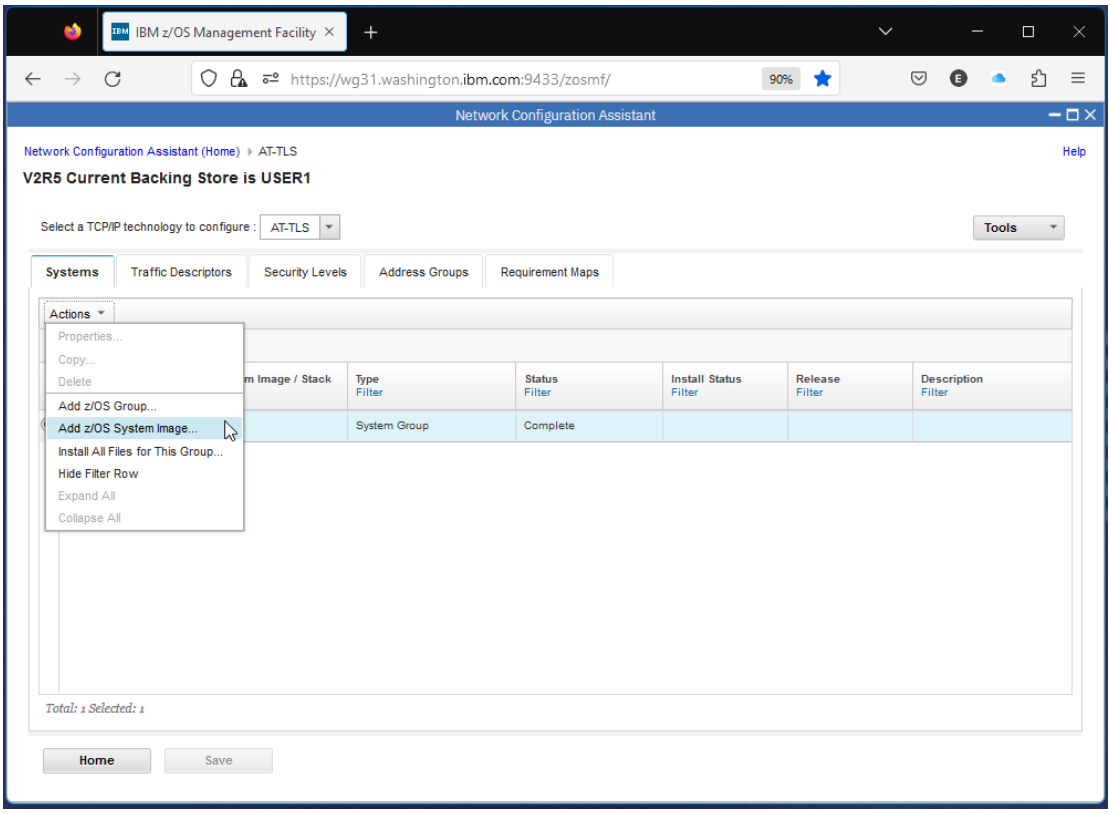
Mitch Johnson (mitchj@us.ibm.com)

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



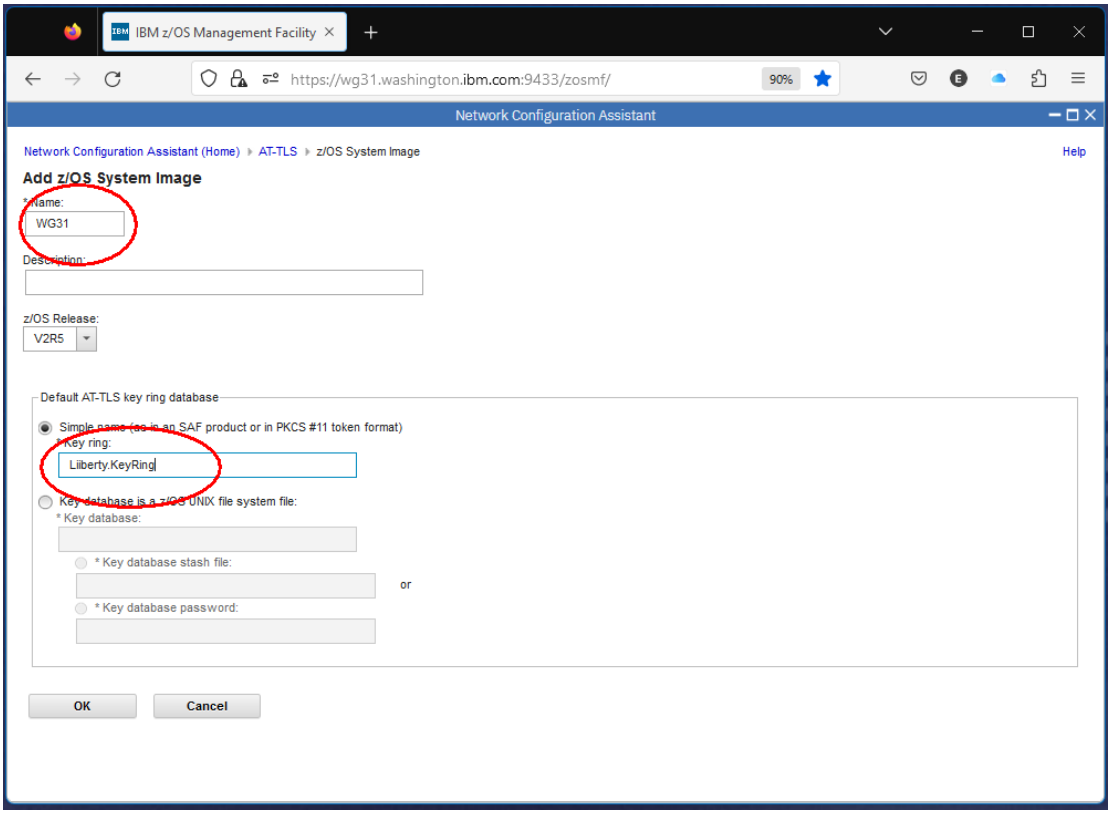
Mitch Johnson (mitchj@us.ibm.com)

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



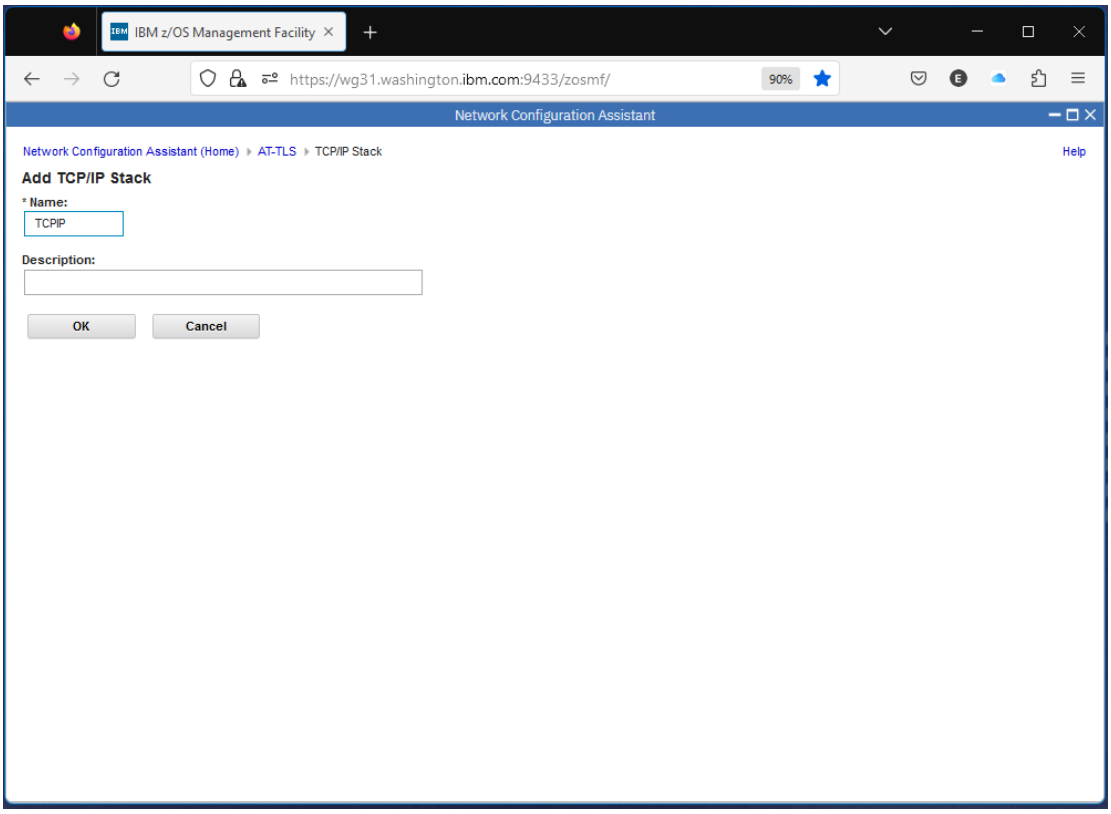
Tech Tip: The value for the key ring will be used if an explicit key ring is not provided for a policy.

We recommend establishing a naming convention for key rings with each SAF identity 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.

Mitch Johnson (mitchj@us.ibm.com)

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



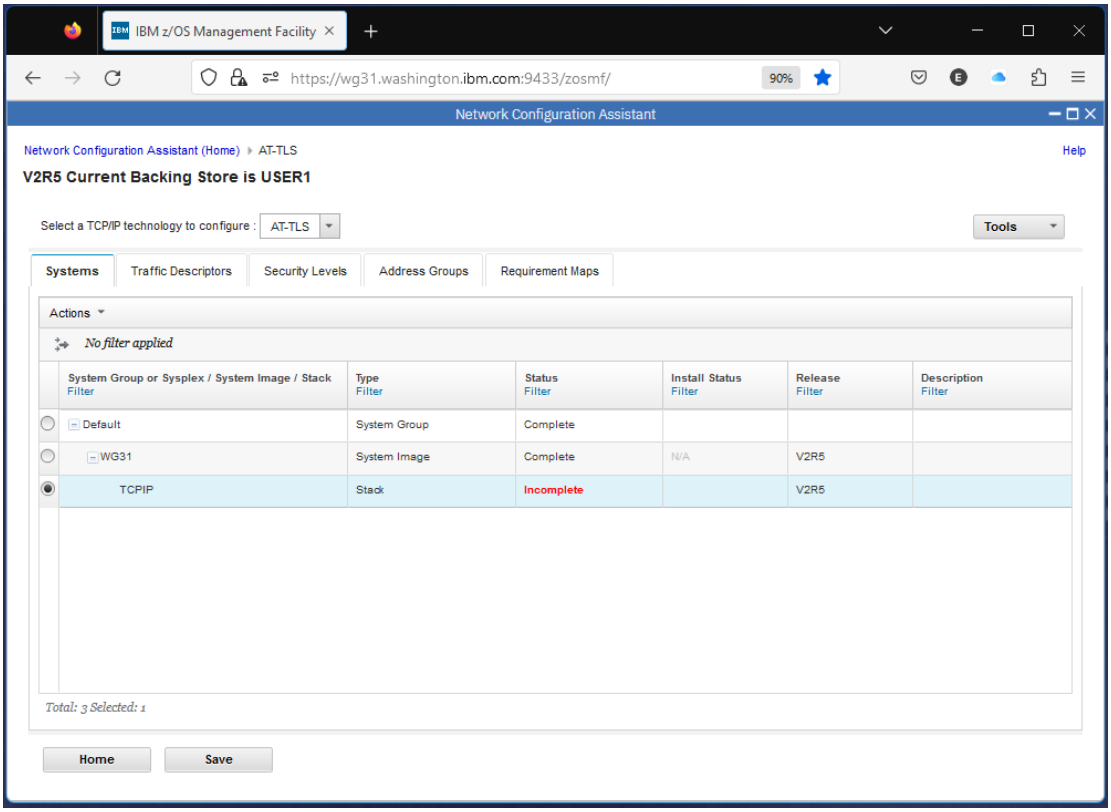
Tech-Tip: The value for the stack name was determined by the TCPIP Name display by entering the MVS command D TCPIP.

```
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
```


Mitch Johnson (mitchj@us.ibm.com)

___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.

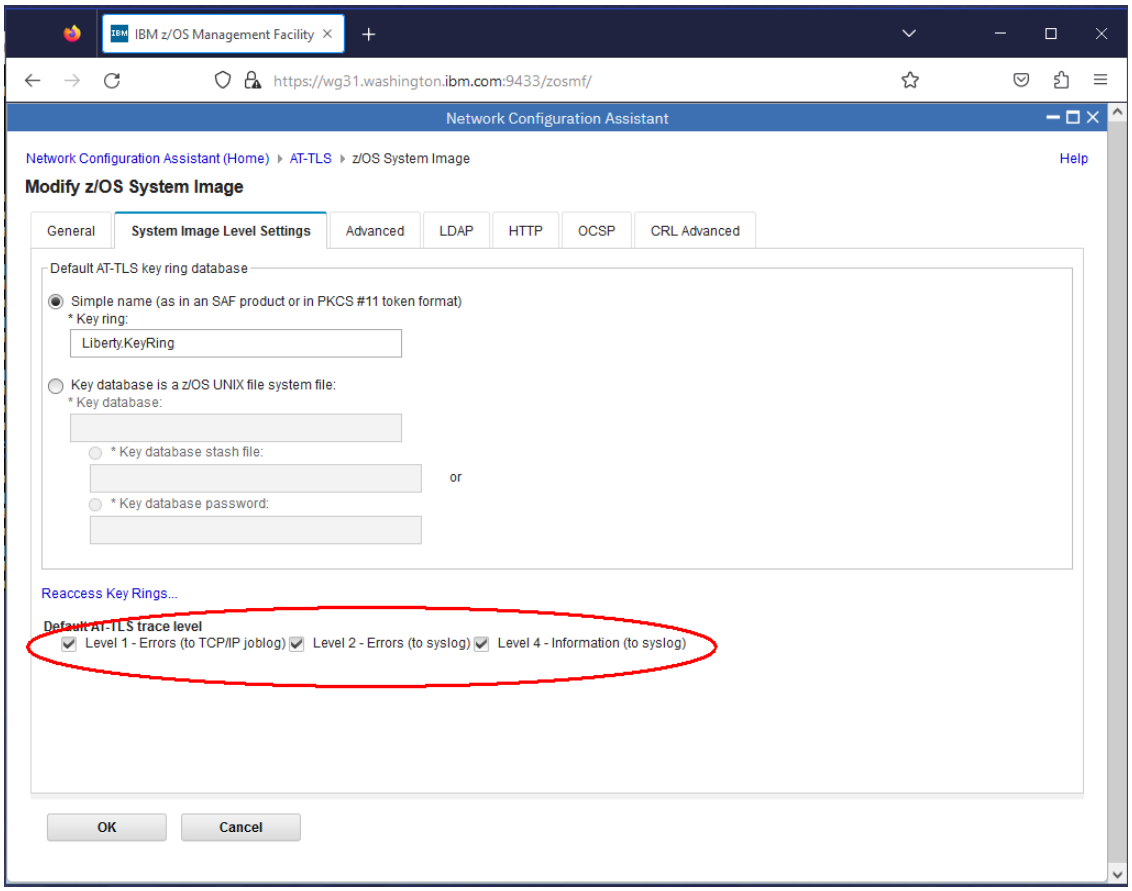
___12. This will display the window below:



Tech Tip: The Incomplete warning will be addressed shortly.

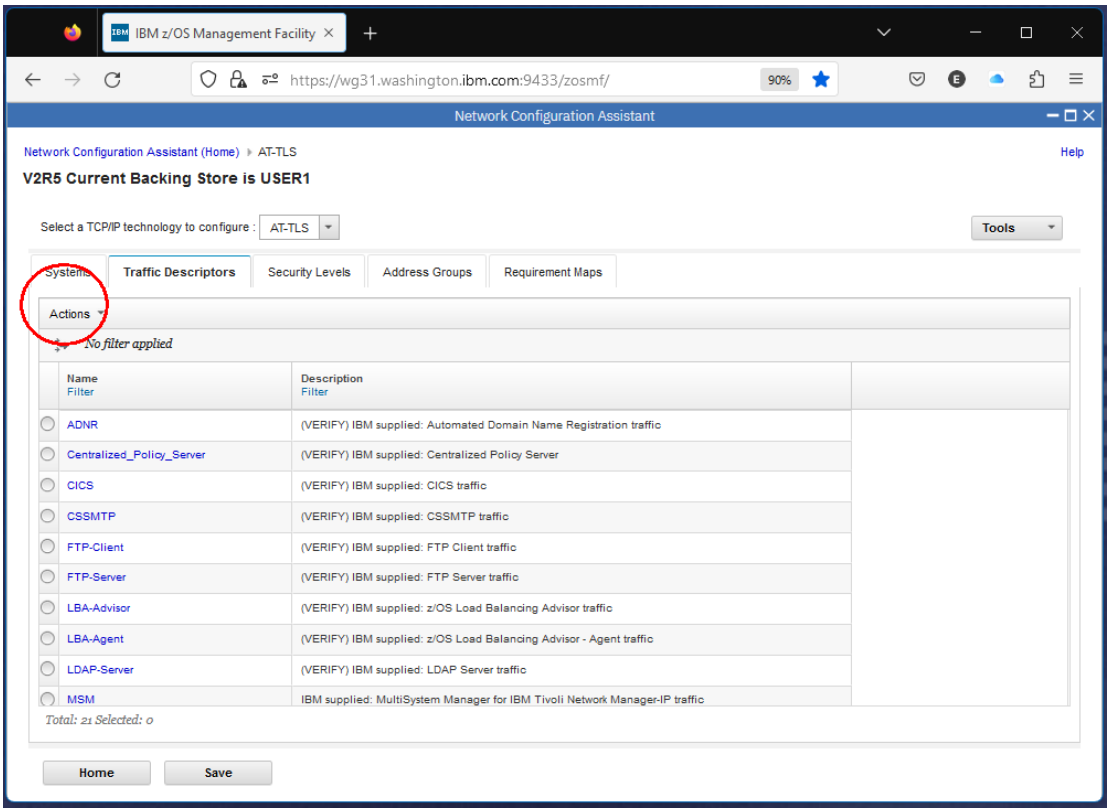
Mitch Johnson (mitchj@us.ibm.com)

___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.



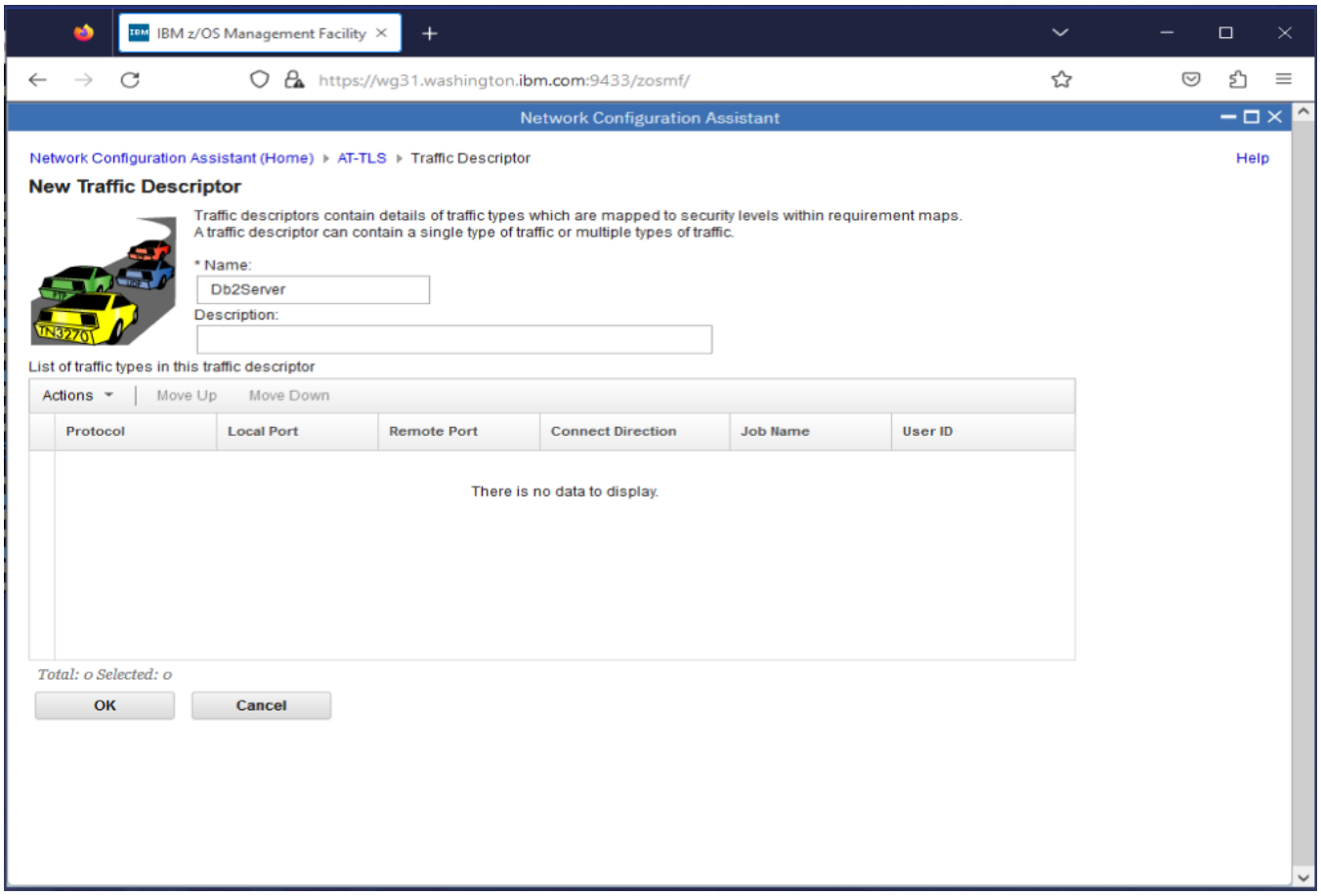
Mitch Johnson (mitchj@us.ibm.com)

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



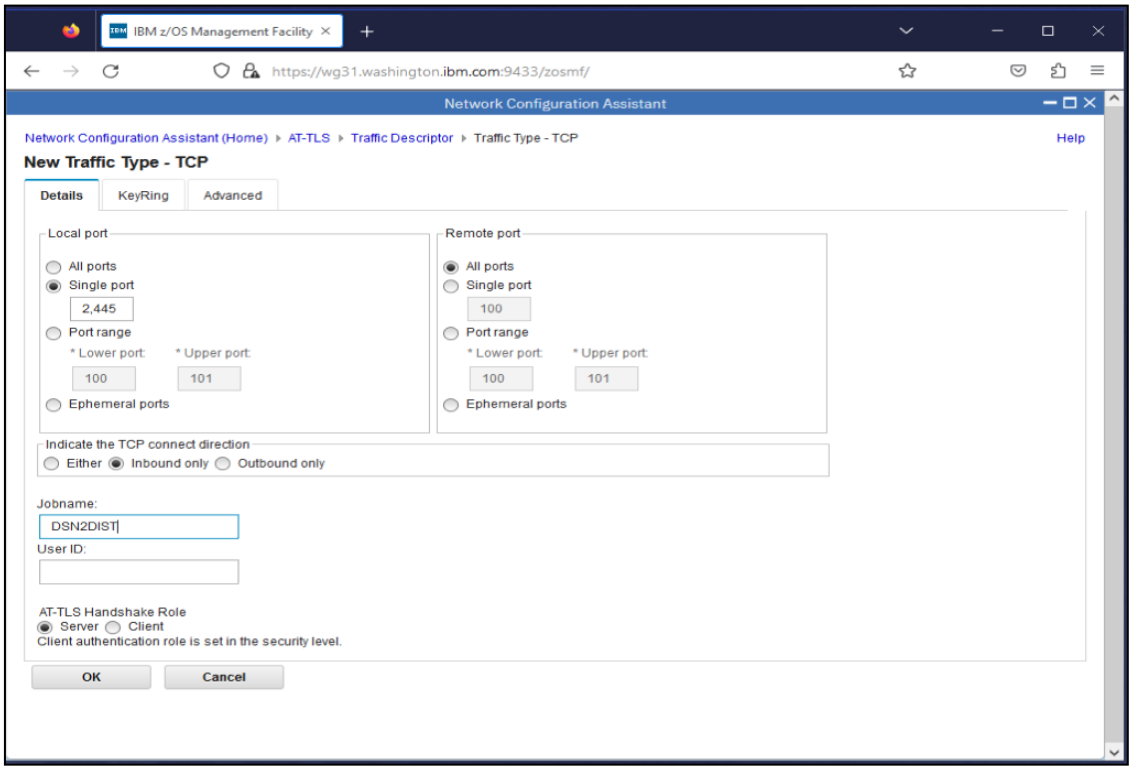
Mitch Johnson (mitchj@us.ibm.com)

___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.



Mitch Johnson (mitchj@us.ibm.com)

___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.



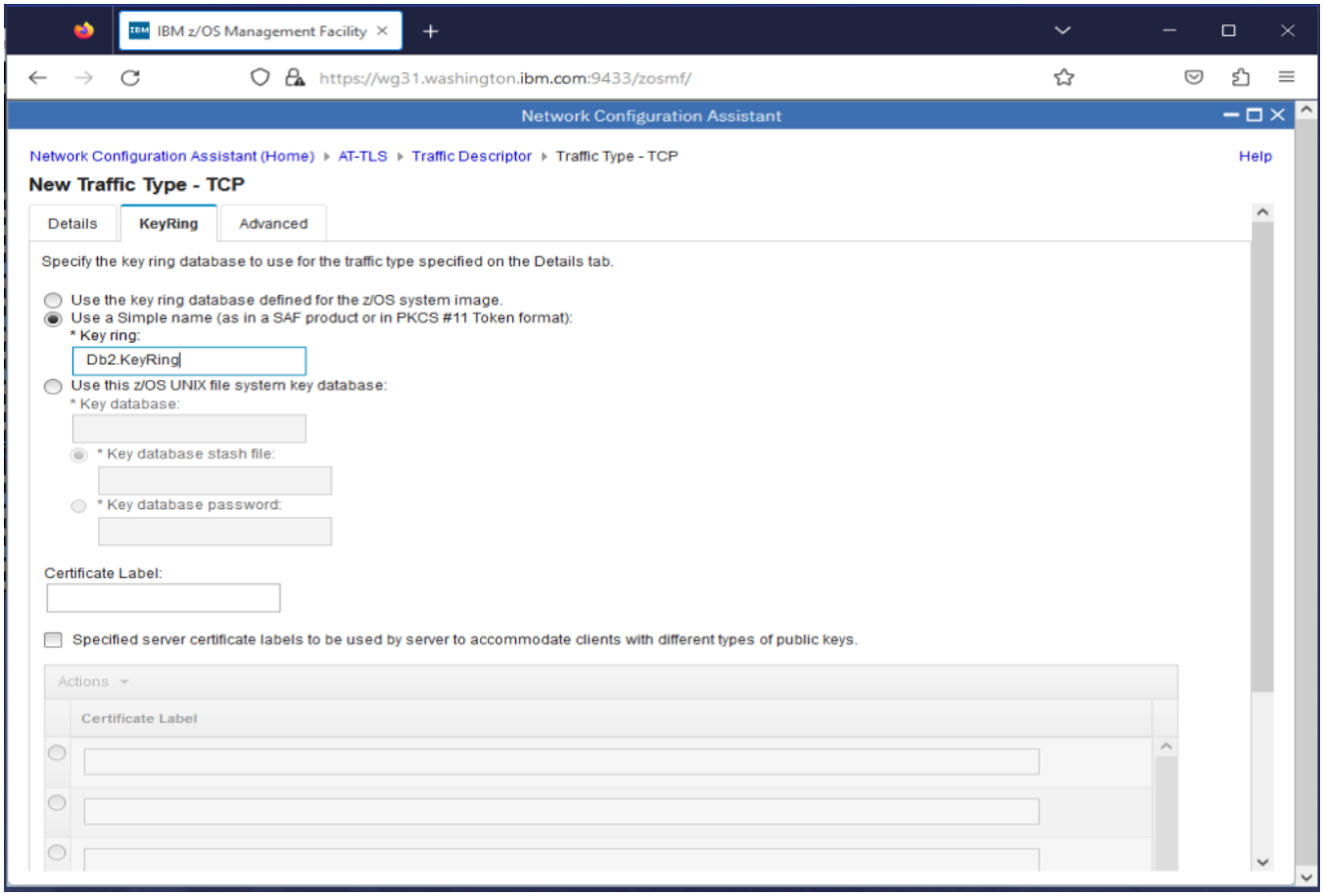
Tech-Tip: This traffic definition is triggered when a client attempts to connect to port *2445*. Port 2445 was identified as the Db2 SECPORT in startup message.

```
DSNL004I  -DSN2 DDF START COMPLETE
          LOCATION  DSN2LOC
          LU        USIBMWZ.DSN2APPL
          GENERICLU -NONE
          DOMAIN    WG31.WASHINGTON.IBM.COM
          TCPPORT   2446
          SECPORT   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.

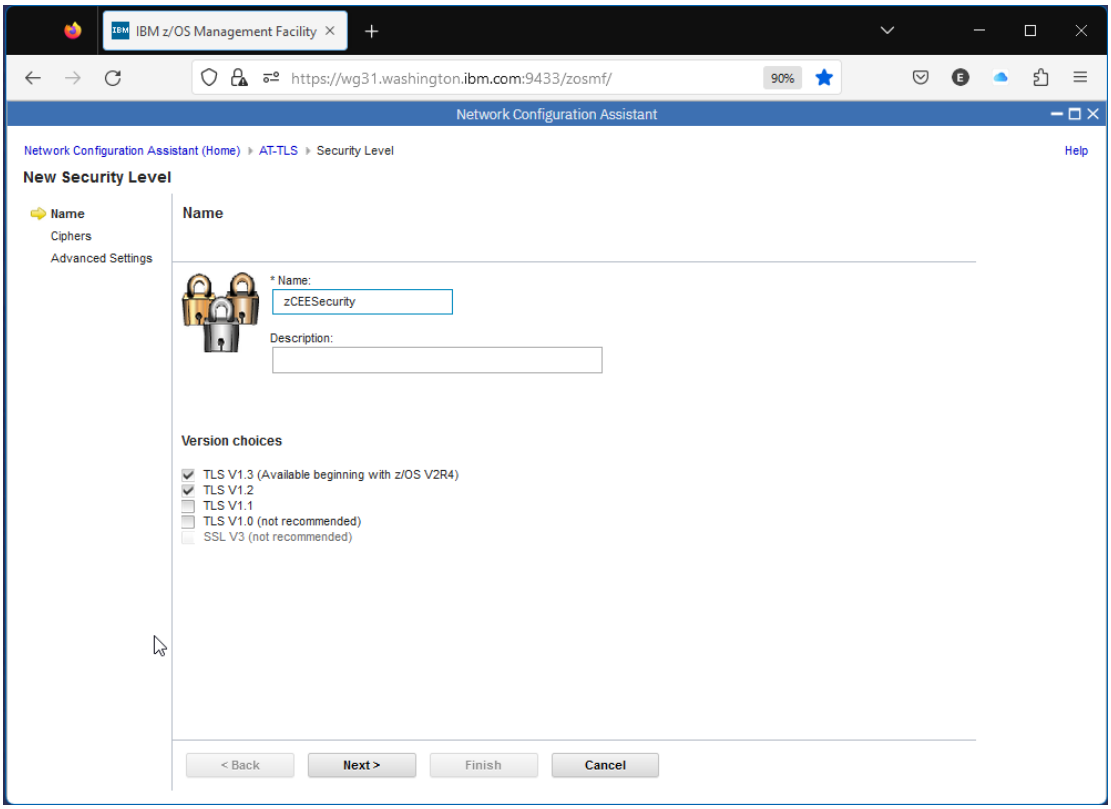
Mitch Johnson (mitchj@us.ibm.com)

___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.



Mitch Johnson (mitchj@us.ibm.com)

___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.

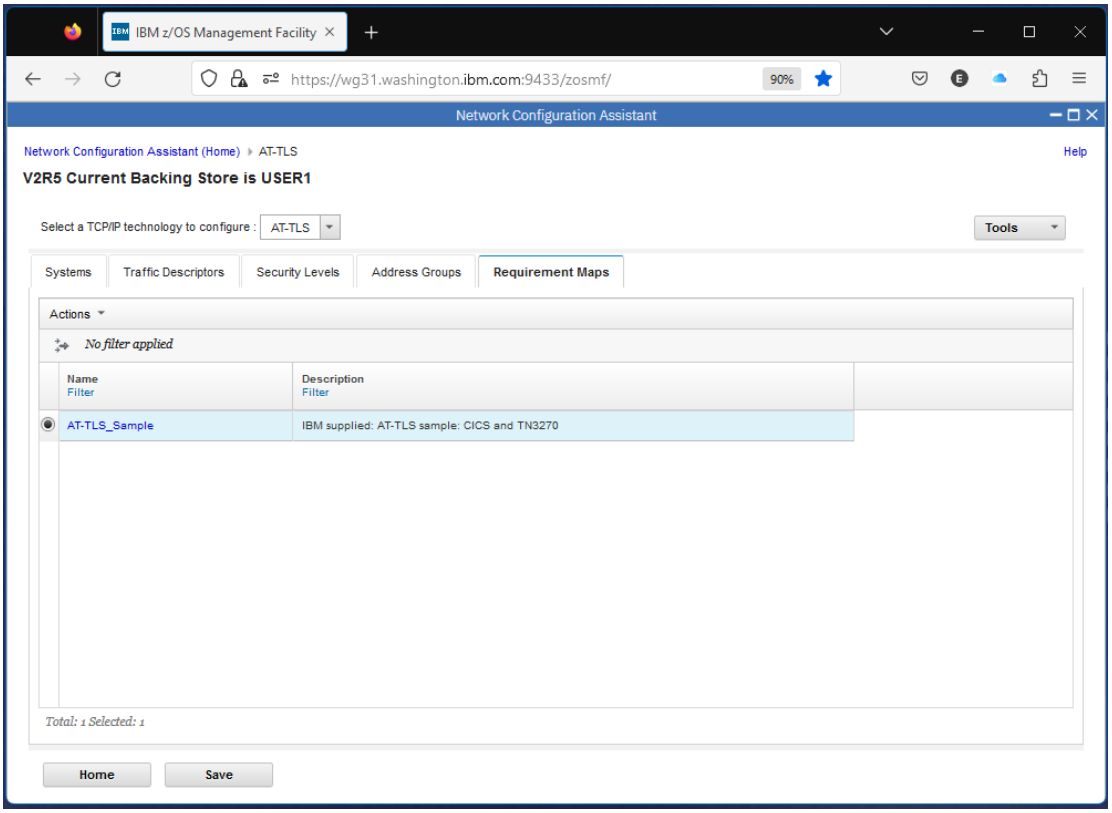
```
     >Db2.KeyRing<
Certificate Label Name              Cert Owner     USAGE      DEFAULT
--------------------------------    ------------   --------   -------
DB2 CA                              CERTAUTH       CERTAUTH     NO
zCEE CA                             CERTAUTH       CERTAUTH     NO
DB2USER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

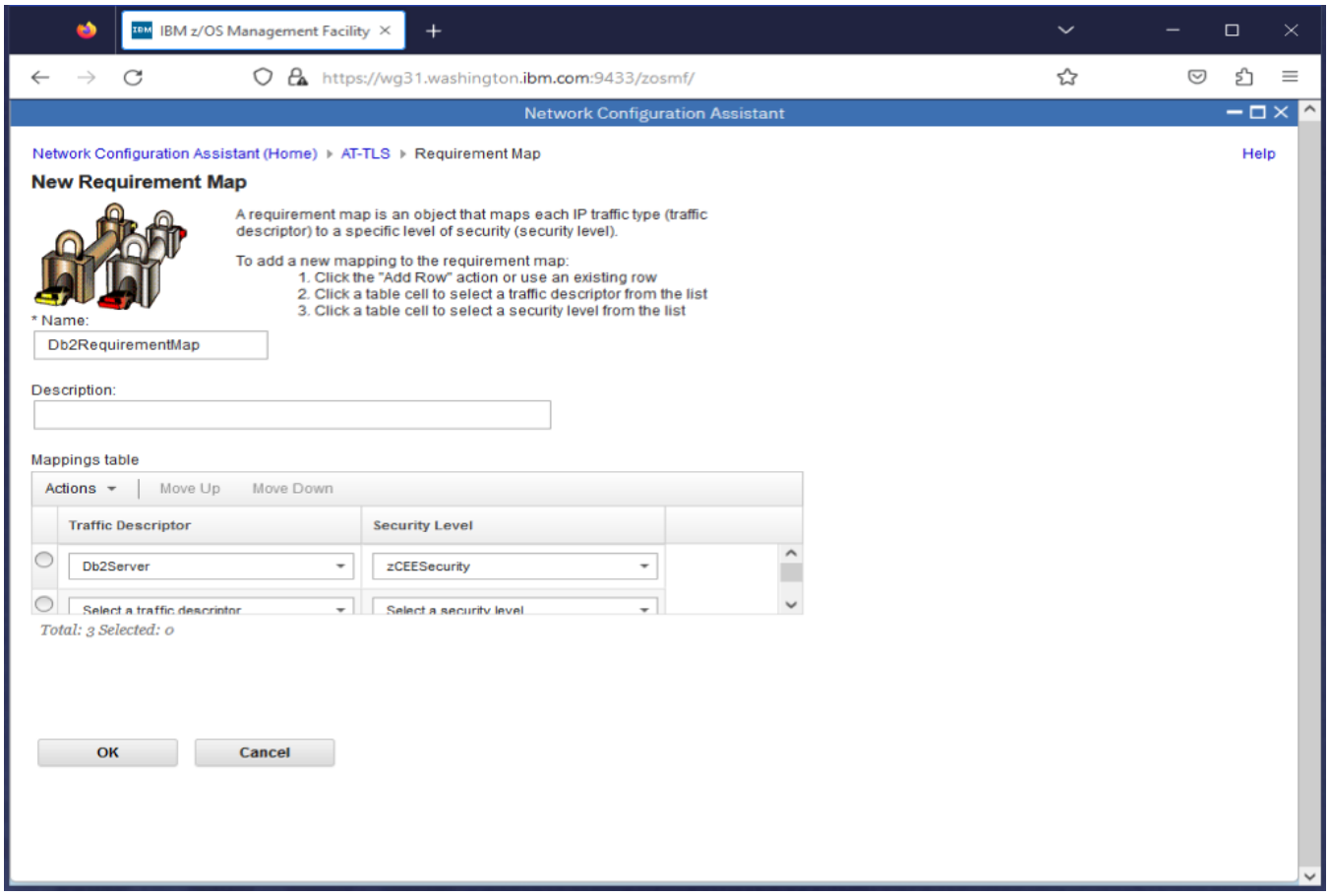

Mitch Johnson (mitchj@us.ibm.com)

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



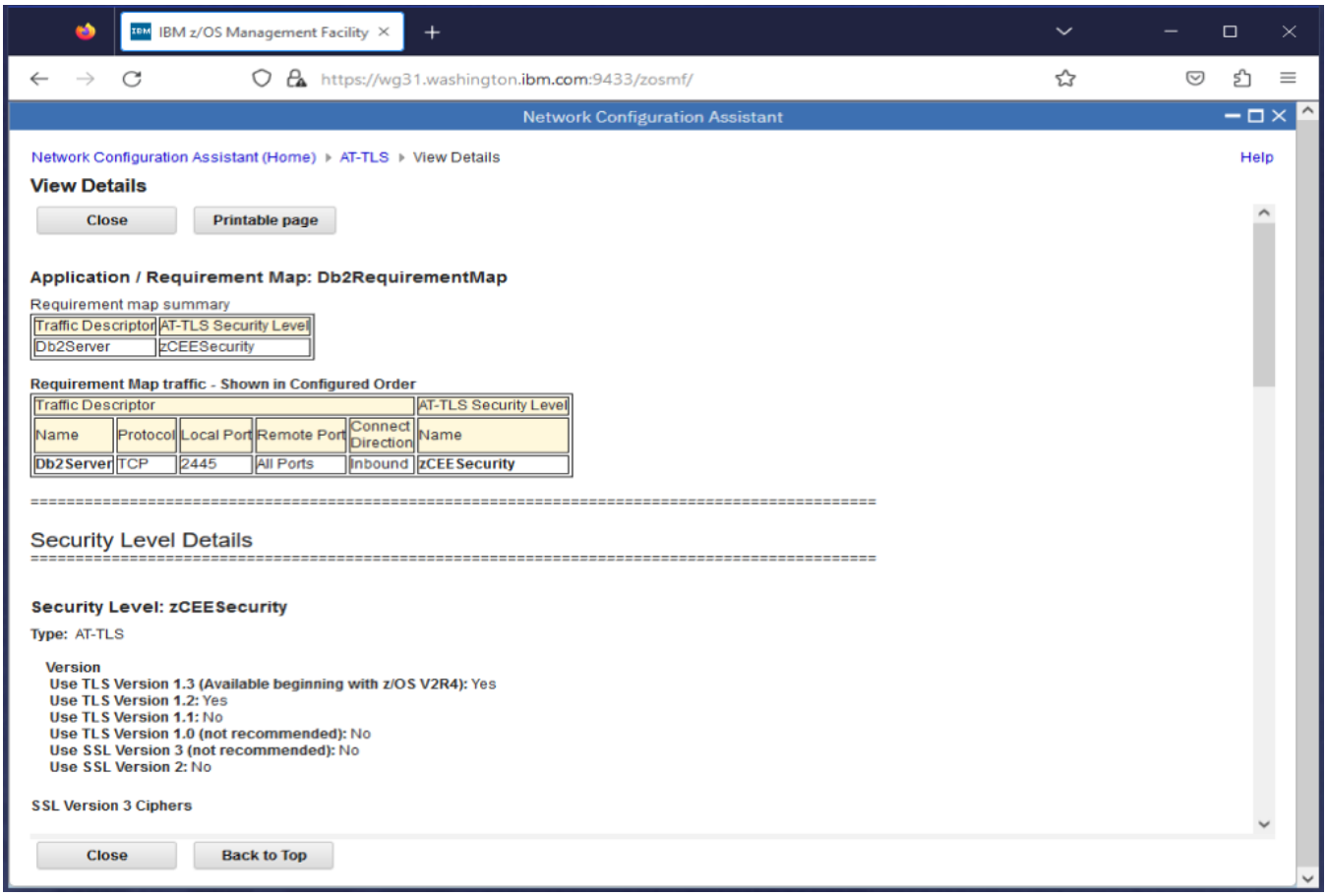
Mitch Johnson (mitchj@us.ibm.com)

___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.



Mitch Johnson (mitchj@us.ibm.com)

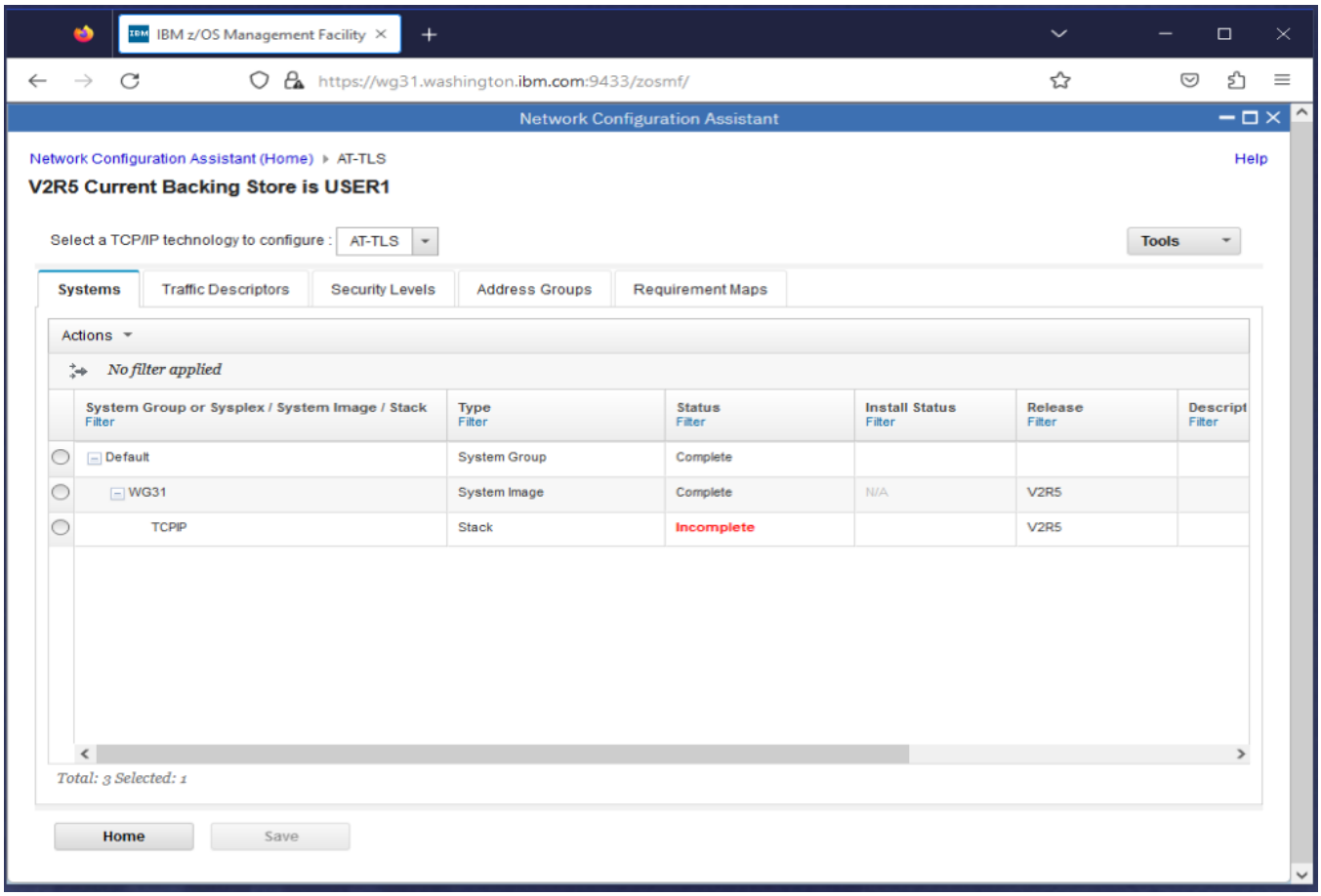
___21. Select the radio button beside *Db2RequirementMap* and use the *Actions* pull-down to select the *View Details* options to display the window below. Review the details and click the **Close** button to continue.



___22. Click the **Save** button to save the configuration.

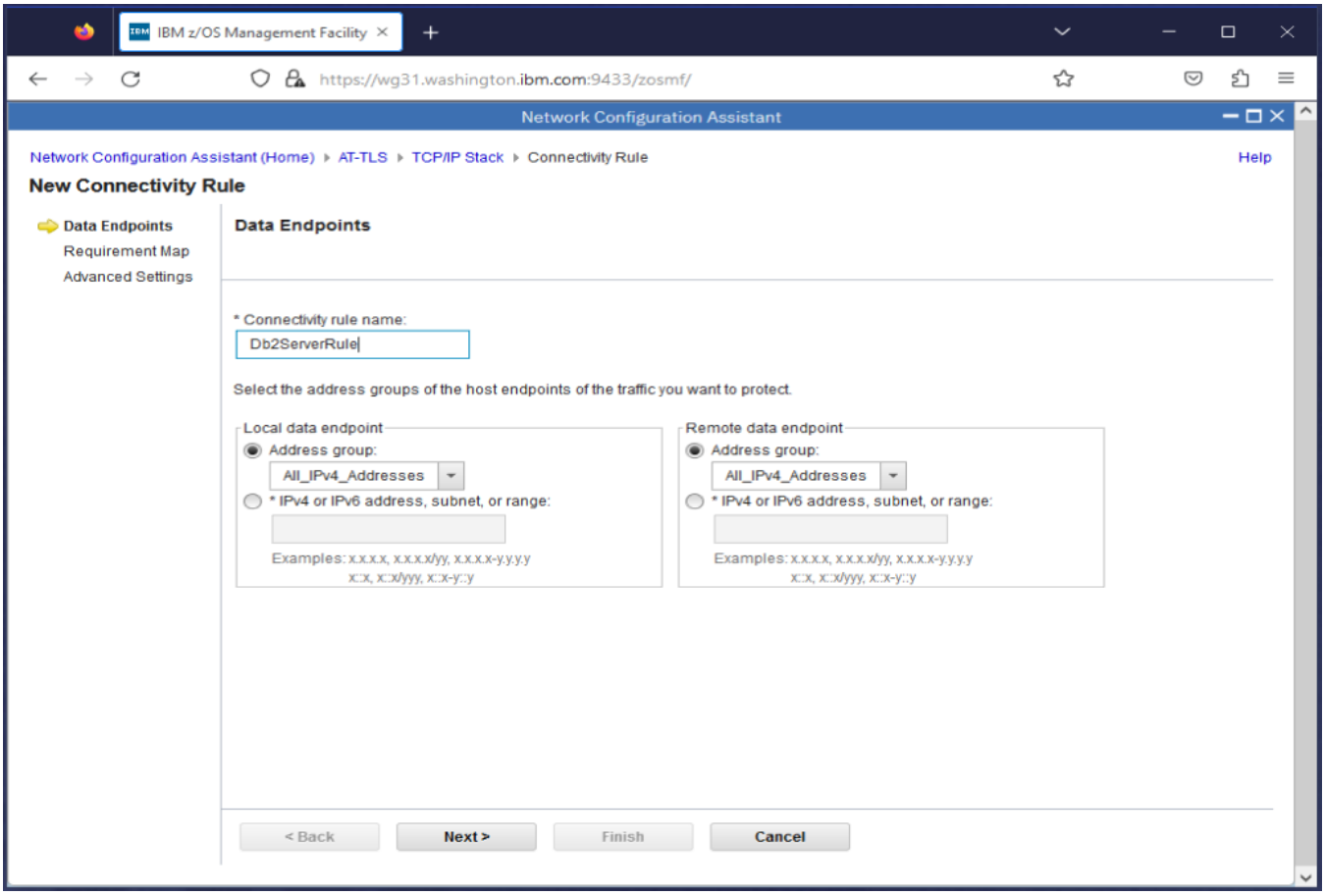
Mitch Johnson (mitchj@us.ibm.com)

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



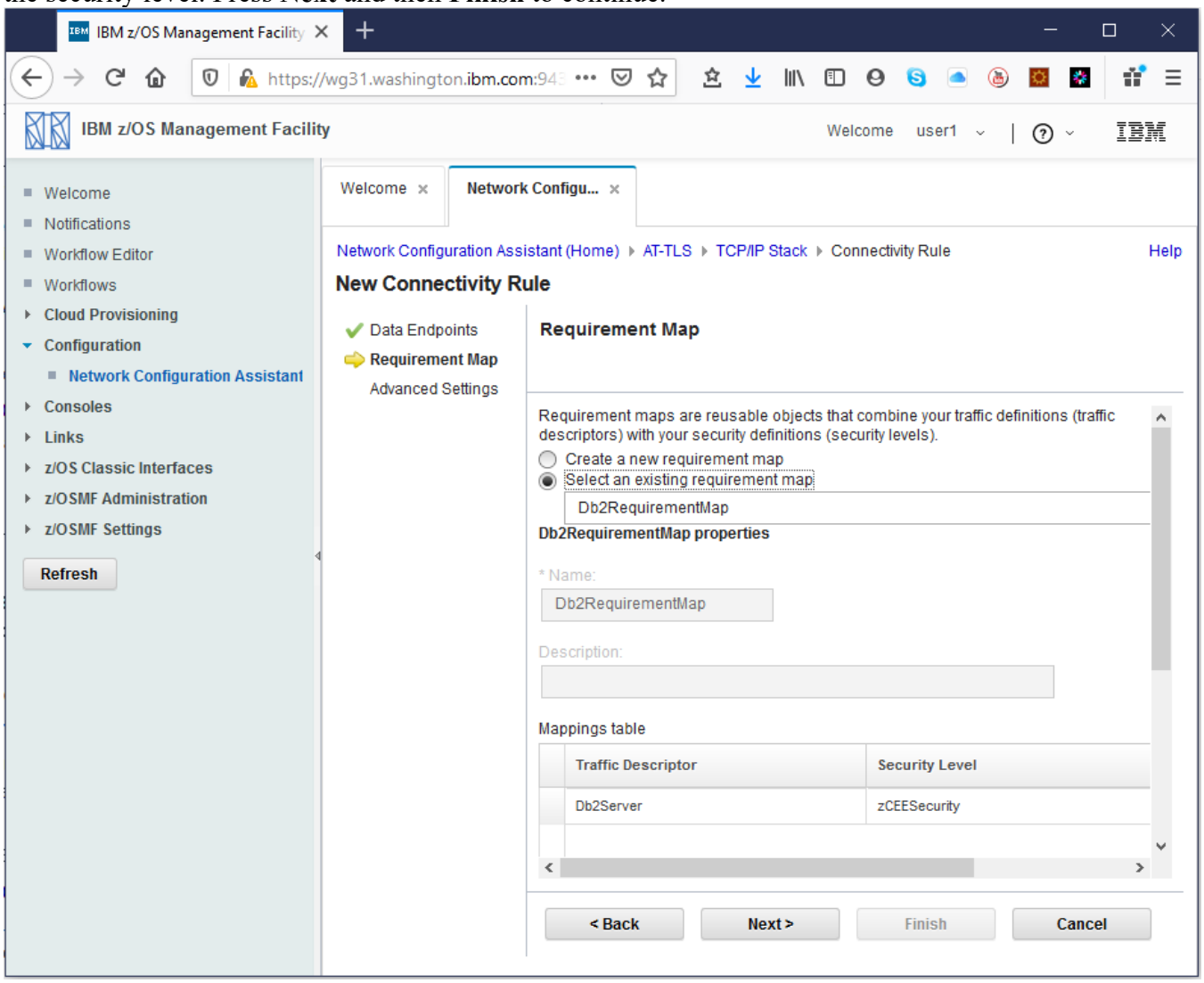
Mitch Johnson (mitchj@us.ibm.com)

___24. Select the radio button beside *TCPIP* 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.



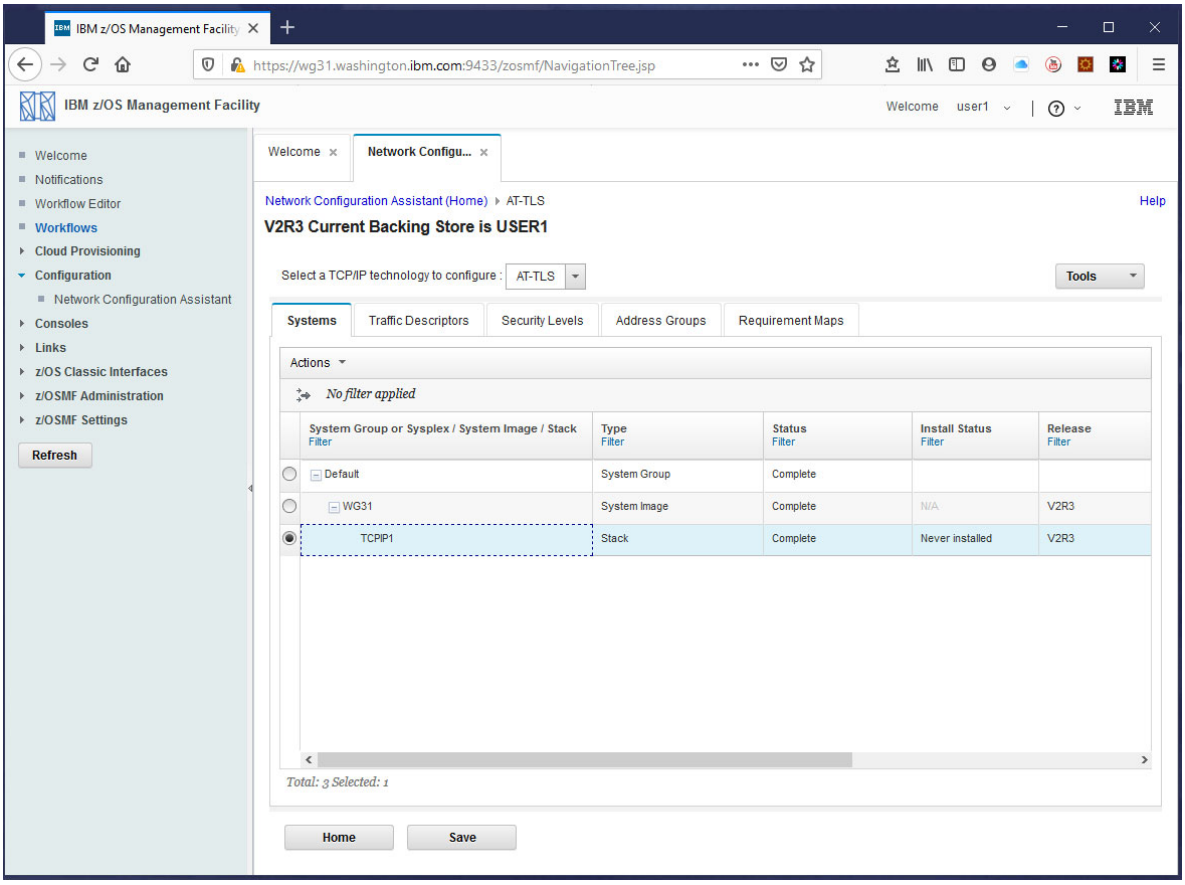
Mitch Johnson (mitchj@us.ibm.com)

___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.



Mitch Johnson (mitchj@us.ibm.com)

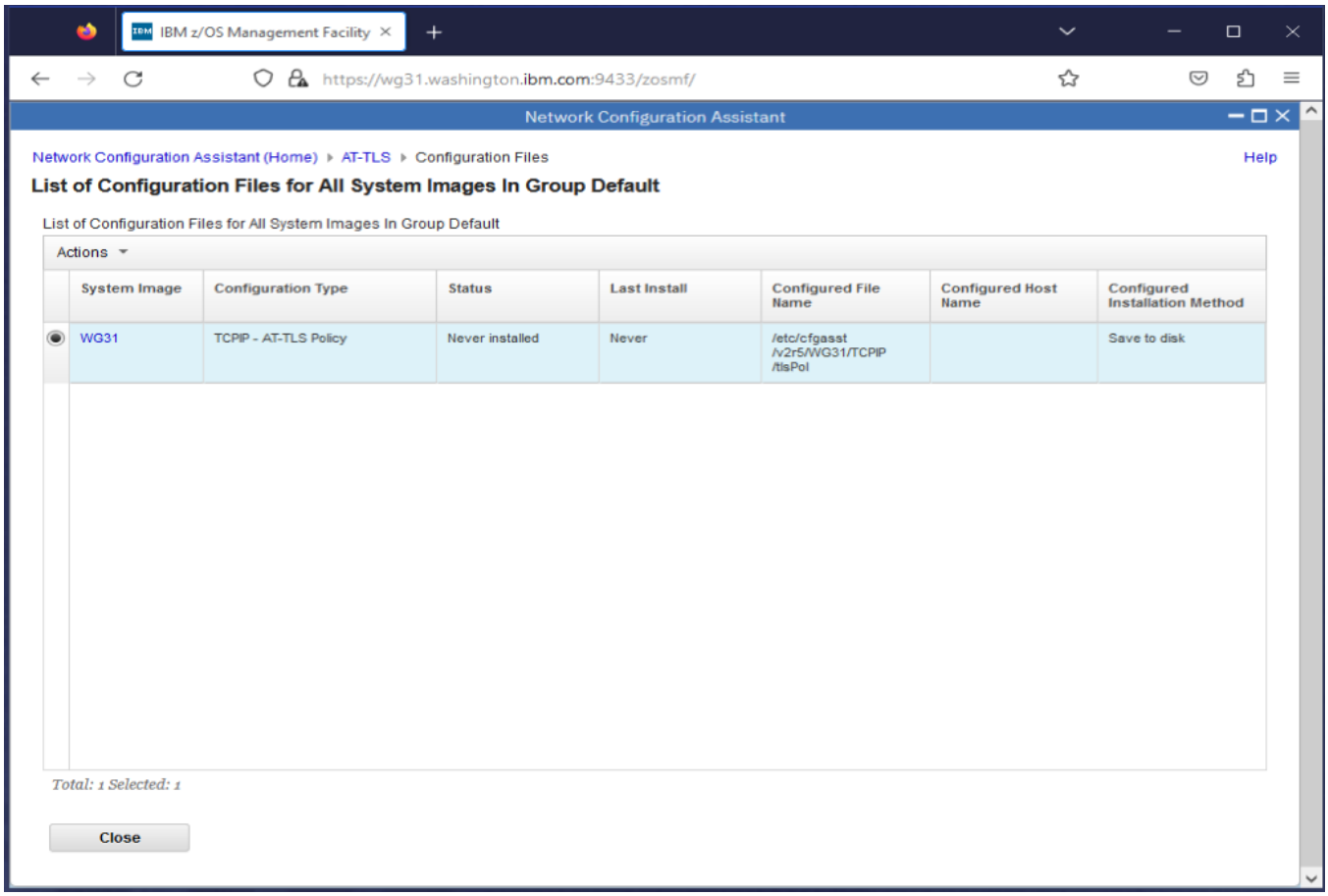
___26. Press **Close** to return to this window. Note that the status of the configuration is now complete (*Never installed).*



___27. Select the radio button beside *TCPIP* and use the *Actions* pull-down to select *Install All Files for This Group.*

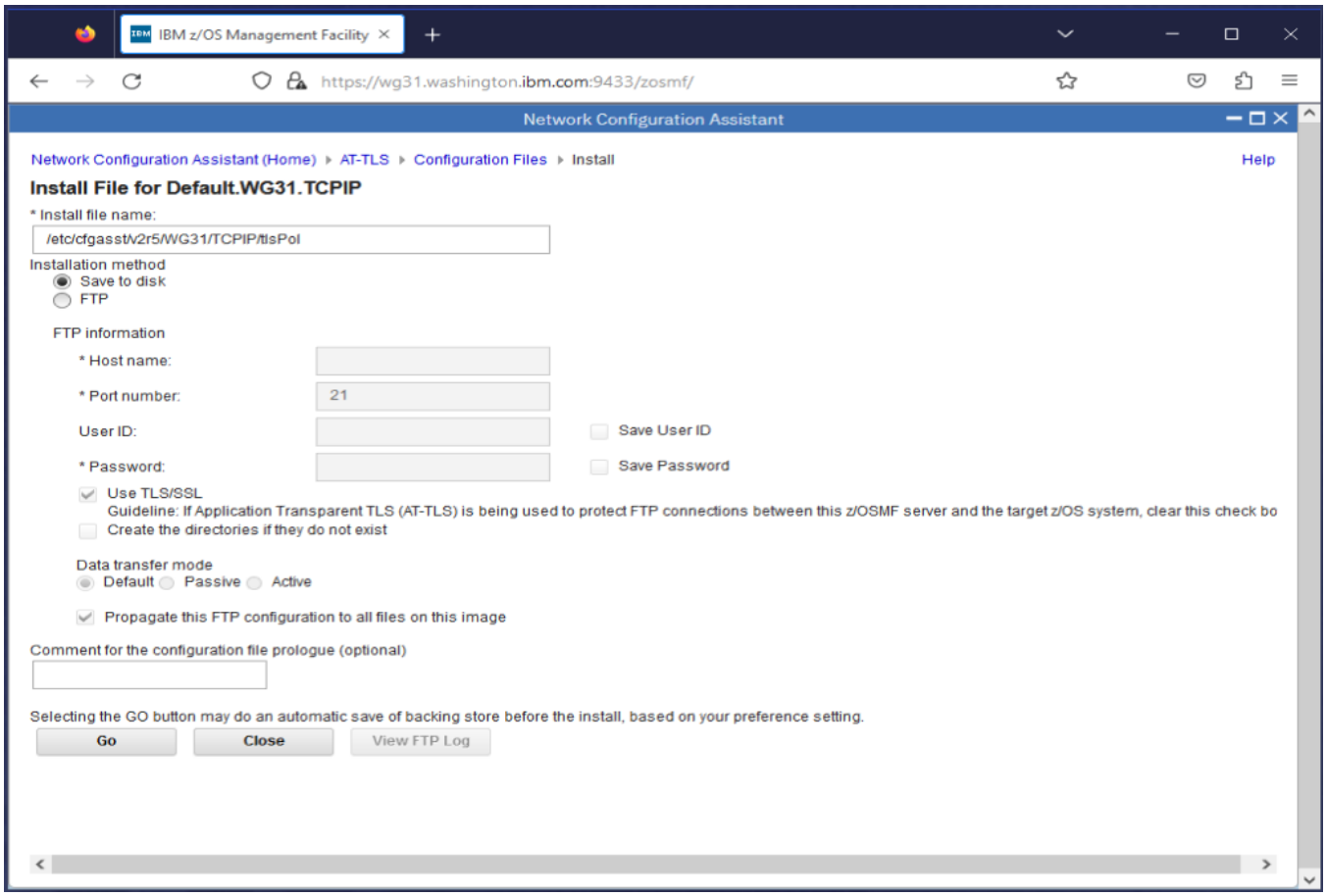
Mitch Johnson (mitchj@us.ibm.com)

___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*.



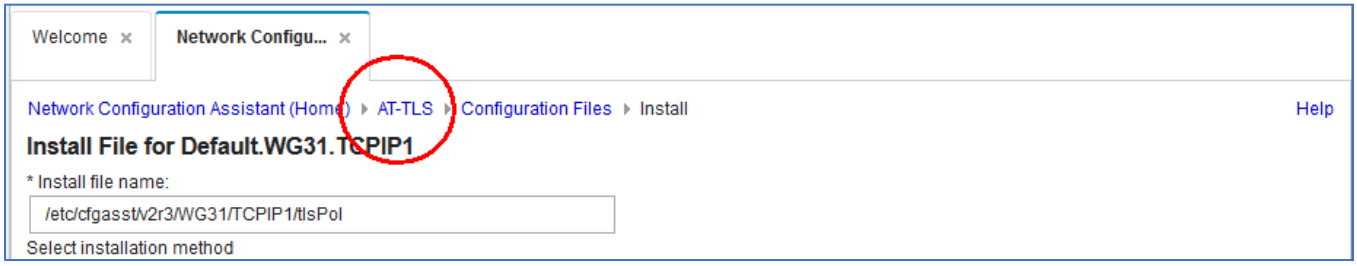
Mitch Johnson (mitchj@us.ibm.com)

___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 it is not active yet.

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).

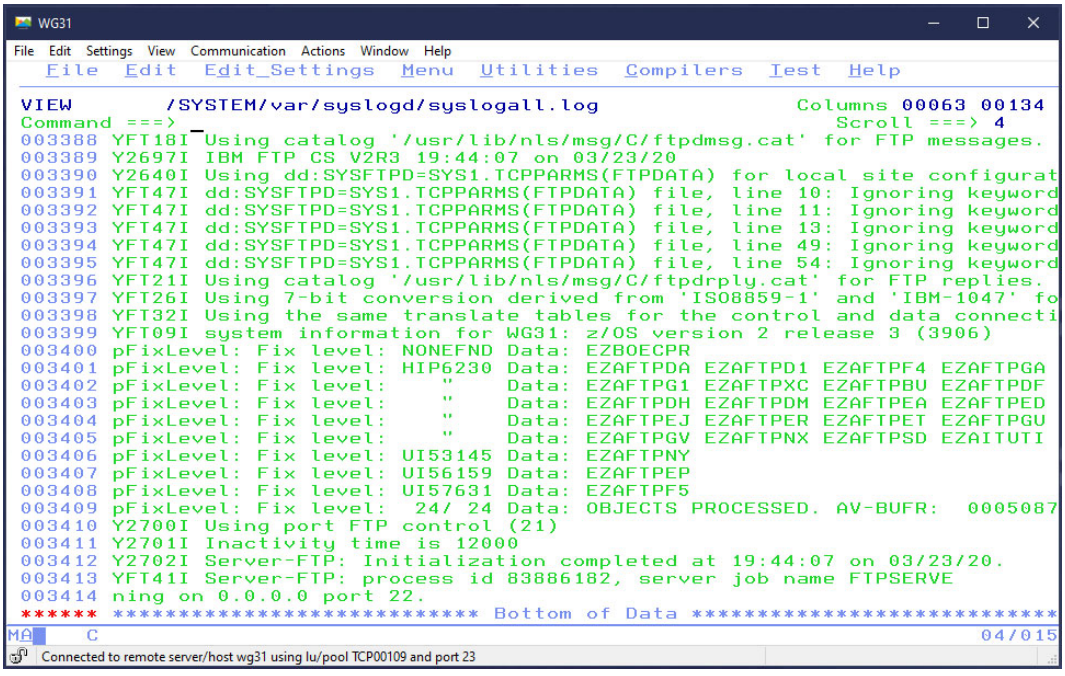
___1. This instance of the policy agent has been configured to use the *SYSLOGD* daemon task to log messages.

```
//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) as shown below:

```
#####################################################################
#
# 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 something like what is shown below:



Mitch Johnson (mitchj@us.ibm.com)

___4. Stop and restart the policy agent task using MVS command, ***P PAGENT*** and ***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 TTLSRule, 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/v2r5/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
```


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. 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, see below:

```
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 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="${shared.config.dir}/db2TLS.xml"/>

```
<include location="${shared.config.dir}/safSecurity.xml"/>
<include location="${shared.config.dir}/ipicIDProp.xml"/>
<include location="${shared.config.dir}/keyringOutboundMutual.xml”/>
<include location="${shared.config.dir}/groupAccess.xml"/>
<include location="${shared.config.dir}/apiRequesterHTTPS.xml"/>
<include location="${shared.config.dir}/imsDatabase.xml"/>
<include location="${shared.config.dir}/db2TLS.xml"/>
<include location="${shared.config.dir}/shared.xml"/>
```


Mitch Johnson (mitchj@us.ibm.com)

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

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

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

___10. Refresh the z/OS Connect server's profile with MVS command ***F BAQSTRT,REFRESH,CONFIG***

Mitch Johnson (mitchj@us.ibm.com)

Test the TLS connection from the z/OS Connect 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

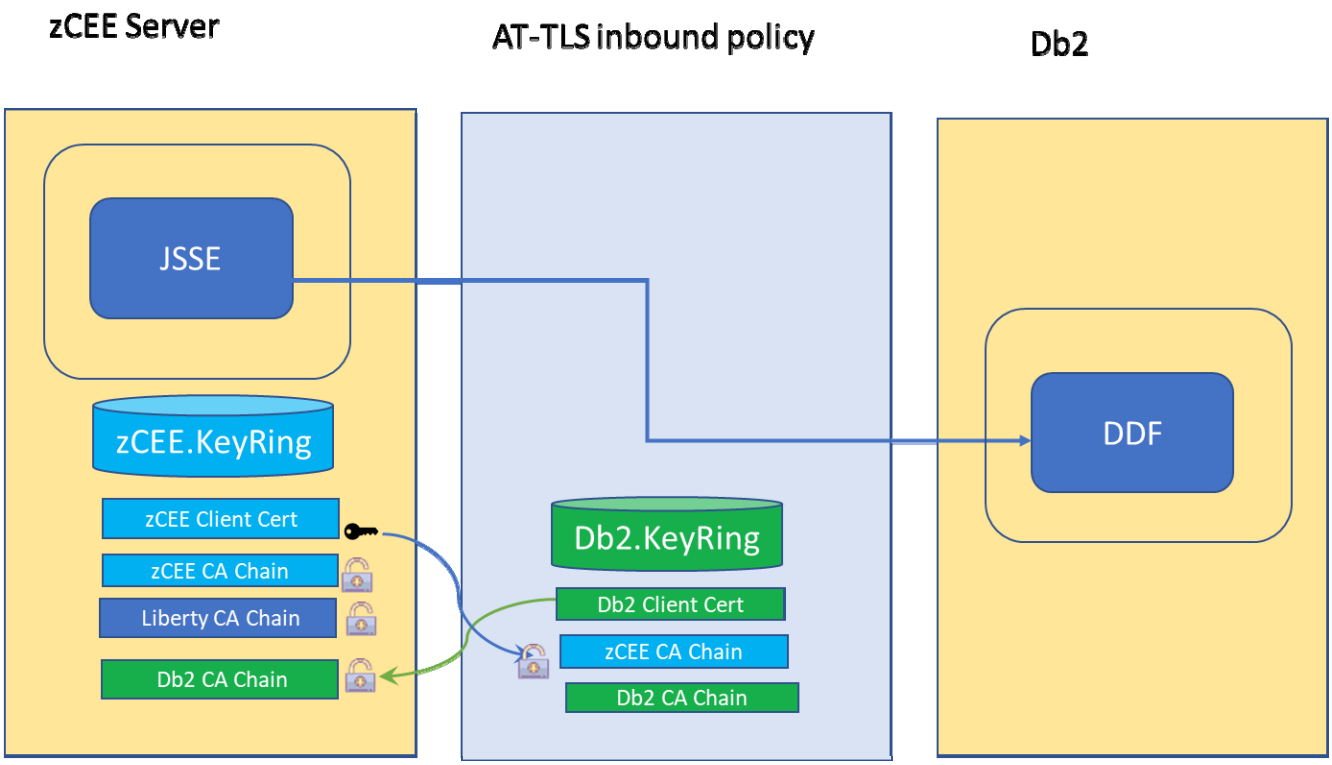
You should see the same results as before. A request from FRED does not work because when the pass ticket for FRED is sent to Db2 this user does not have sufficient authority. When the pass ticket for USER1 is sent to DB2 this user has sufficient authority to access Db2.

___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
```


Mitch Johnson (mitchj@us.ibm.com)

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 EZD1281 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

Mitch Johnson (mitchj@us.ibm.com)

___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_

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
}
```


Mitch Johnson (mitchj@us.ibm.com)

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


Mitch Johnson (mitchj@us.ibm.com)