IBM z/OS Connect EE V3.0

# Customization - Security and IMS Database


IBM Z
Wildfire Team –
Washington System Center

*Lab Version Date: May 29, 2020*

# Table of Contents

**Overview** ..... **4**
***Enabling RACF Pass Tickets*** ..... **5**
*Test with RACF Pass Tickets not enabled* ..... *5*
*Test with identity propagation enabled using RACF Pass Tickets* ..... *7*
*Summary* ..... *9*
***Configuring TLS security to an IMS Subsystem*** ..... **10**
*Creating IMS SAF resources* ..... *10*
*Configure the AT-TLS policies* ..... *16*
***Activating the AT-TLS configuration*** ..... **48**
*Test the TLS connection from the zCEE Server to IMS* ..... *51*
*Optional* ..... *53*
**Summary** ..... **54**

**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 *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.



Mitch Johnson (mitchj@us.ibm.com)

# Overview

This exercise demonstrates the steps required to enable security between a z/OS Connect EE (zCEE) server and IMS.

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 IMS Connect and this identity will subsequently be used for IMS authorization checks.

In part two of the exercise, TLS support will be added by configuring AT-TLS policies. The presence of these policies will act as a surrogate for handing the server role on behalf of IMS Connect and a surrogate for the z/OS Connect server as client.



Mitch Johnson (mitchj@us.ibm.com)

# *Enabling RACF Pass Tickets*

When sending request from an z/OS Connect server to IMS the identity used for IMS authorization checks will be by default the identity configured in the basic authentication element of the IMS connection factory. This identity is associated with the server and not the z/OS Connect authenticated identity under which the requests is running.

To provide identity assertion of the authenticated identity the use of RACF Pass Tickets is required. When RACF Pass Tickets are enabled, the zCEE server will obtain a pass ticket from RACF and send this ticket (token) along with the request to IMS Connect. When the request arrives at the IMS Connect the pass ticket will be validated with RACF and the z/OS Connect authenticated identity will be extracted and use for subsequent IMS authorization checks.

## *Test with RACF Pass Tickets not enabled*

But first let's explore invoking an IMS API and observing the security 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 *imsDatabase.xml* (if they are not already present) see below:

***<include location="${server.config.dir}/includes/imsDatabase.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="${server.config.dir}/includes/imsDatabase.xml"/>
<include location="${server.config.dir}/includes/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 a MVS command and you may enter the command at the prompt by using the / (slash) prefix or using the *SDSF – System Command Extension* panel.



Mitch Johnson (mitchj@us.ibm.com)

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

___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/imdb/contact/LAST3***

```
c:\z\admin>curl -X get --cacert certauth.pem --cert fred.p12:secret --cert-type P12
https://wg31.washington.ibm.com:9443/imdb/contact/LAST3
{"response":{"result":[{"firstName":"FIRST3","zipCode":"D03\/R03","lastName":"LAST3","
phoneNumber":"8-111-3333"}]}}}
```

___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/imdb/contact/LAST3***

You should see the same results because the requests to IMS Connect are using the basic authentication as configured in the connection factory in *imsDatabase.xml* (see below).

```
<connectionFactory id="DFSIVPAConn">
<properties.imsudbJLocal
   databaseName="DFSIVPA"
   datastoreName="IVP1"
   datastoreServer="wg31.washington.ibm.com"
   driverType="4"
   portNumber="5555"
   user="USER1"
   password="USER1"
   flattenTables="True"/>
</connectionFactory>
```

___6. To confirm that USER1 is being used, locate the outstanding reply for IMS Connect in the SDSF log and respond with ***VIEWHWS***, as in ##VIEWHWS (where ## is the reply number). You should see two requests for port 5555 both with the same *CLIENTID* of *USER1*.

```
HWSC0001I    PORT=5555D    STATUS=ACTIVE     KEEPAV=0 NUMSOC=3
EDIT=         TIMEOUT=6000
HWSC0001I      CLIENTID USERID   TRANCODE DATASTORE STATUS       SECOND
CLNTPORT IP-ADDRESS            APSB-TOKEN
HWSC0001I      ODB970D4 USER1              IVP15OOD  RECV             44
1423   192.168.017.220
HWSC0001I      ODB36ED4 USER1              IVP15OOD  RECV            153
1349   192.168.017.220
HWSC0001I      TOTAL CLIENTS=2  RECV=2 READ=0 CONN=0 XMIT=0 OTHER=0
```



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 submit the job in member *IMSPASS* in data set *USER1.ZCEE30.CNTL*.

```
ADDGROUP ZCEEIMS

CONNECT LIBSERV GROUP(ZCEEIMS)

SETROPTS CLASSACT(PTKTDATA) RACLIST(PTKTDATA)
SETROPTS GENERIC(PTKTDATA)

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

RDEFINE PTKTDATA IRRPTAUTH.IMSAPPL.* UACC(NONE)
PERMIT IRRPTAUTH.IMSAPPL.* ID(ZCEEIMS) CLASS(PTKTDATA) ACC(UPDATE)

SETROPTS RACLIST(PTKTDATA) REFRESH
```

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

**Tech-Tip:** The value *IMSAPPL* was derived from the IMS Connect *APPL* attribute for the ODACCESS configuration entry, see sample below:

HWS=(ID=IMS15HWS,XIBAREA=100,RACF=Y,RRS=Y)
TCPIP=(HOSTNAME=TCPIP,PORTID=(4000,LOCAL),RACFID=JOHNSON,TIMEOUT=5000)
DATASTORE=(GROUP=OTMAGRP,ID=IVP1,MEMBER=HWSMEM,DRU=HWSYDRU0,
TMEMBER=OTMAMEM,APPL=IMSAPPL)
ODACCESS=(ODBMAUTOCONN=Y,IMSPLEX=(MEMBER=IMS15HWS,TMEMBER=PLEX1),
DRDAPORT=(ID=5555,PORTTMOT=6000),ODBMTMOT=6000,APPL=**IMSAPPL**)

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 *imsdatabase.xml* to an include for ***imsDatabasePassTickett.xml***.

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

The contents of imsDatabasePassTicket.xml are shown below

```
<connectionFactory id="DFSIVPAConn">
<properties.imsudbJLocal
   databaseName="DFSIVPA"
   datastoreName="IVP1"
   datastoreServer="wg31.washington.ibm.com"
   driverType="4"
   portNumber="5555"
   applicationName="IMSAPPL"
   flattenTables="True"/>
</connectionFactory>
```

___3. Refresh the server's configuration using the MVS command ***F BAQSTRT,REFRESH,CONFIG***

**Tech-Tip:** A refresh should be sufficient, but a complete restart of the server might be better. The TCPIP connects linger for a while and restart of the server will break these connections and provide distinct results between tests.

___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/imdb/contact/LAST3***

```
c:\z\admin>curl -X get --cacert certauth.pem --cert fred.p12:secret --cert-type P12
https://wg31.washington.ibm.com:9443/imdb/contact/LAST3
{"response":{"result":[{"firstName":"FIRST3","zipCode":"D03\/R03","lastName":"LAST3","
phoneNumber":"8-111-3333"}]}}
```

:

You should see the same results.



Mitch Johnson (mitchj@us.ibm.com)

___5. To confirm the FRED identity (fred.p12) is being used, locate the outstanding reply for IMS Connect in the SDSF log again and respond with IMS Connect command *VIEWHWS*, as in ***##VIEWHWS*** (where ## is the outstanding reply number for IMS Connect). You should see two requests for port 5555, with one *CLIENTID* for FRED and another *CLIENTID* for USER1.

```
HWSC0001I    PORT=5555D    STATUS=ACTIVE     KEEPAV=0 NUMSOC=3
EDIT=         TIMEOUT=6000
HWSC0001I      CLIENTID USERID   TRANCODE DATASTORE STATUS       SECOND
CLNTPORT IP-ADDRESS            APSB-TOKEN
HWSC0001I      ODBC70D1 FRED               IVP15OOD  RECV             23
1547   192.168.017.220
HWSC0001I      ODBA4CD5 USER1              IVP15OOD  RECV            251
1523   192.168.017.220
HWSC0001I      TOTAL CLIENTS=2  RECV=2 READ=0 CONN=0 XMIT=0 OTHER=0
```

.

## *Summary*

In this section a simple REST client (cURL) has been used to invoke and API which accesses IMS. The API was initially tested without using RACF Pass Tickets using basic authentication from 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 IMS Connect. Finally, the REST client was used to demonstrate that the identity associated with the client certificates (fred.p12 and user1.p12) were propagated to IMS Connect for authorization checks.



Mitch Johnson (mitchj@us.ibm.com)

# *Configuring TLS security to an IMS Subsystem*

Adding TLS support to the connection between the z/OS Connect server and IMS requires the creation of a key ring belonging to the identity under which the IMS Connect task is executing (look for message IEF695I in the IMS15HWS task's JES messages). This key ring contains the personal and 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 IMS SAF resources*

First, we may have to do some housekeeping depending on which exercises have been previously performed.

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

___2. Browse member **ZCEERCF7**. You should see the RACF commands below. Submit the job for execution if this 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)
```

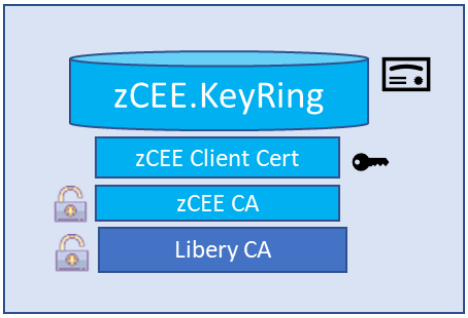


Mitch Johnson (mitchj@us.ibm.com)

These commands

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

Below is visual representation of the key ring just created




___3. Edit the *server.xml* configuration file for the *myServer* server, located in diretory */var/zosconnect/servers/myServer* 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"/>***



Mitch Johnson (mitchj@us.ibm.com)

___4. Next, browse member **IMSTLS**, you should see the RACF commands below. Submit the job for execution.

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

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

setr raclist(digtcert,digtnmap) refresh

 /* Create IMS key ring and connect CA and personal certificates */
racdcert id(IMSSTC) addring(IMS.KeyRing)

racdcert id(IMSSTC) connect(ring(IMS.KeyRing) +
          label('IMS CA') certauth usage(certauth))

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

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

 /* Connect default personal certificiate */
racdcert id(IMSSTC) connect(ring(IMS.KeyRing) +
          label('IMSSTC') default

setropts raclist(digtring,digtnmap) refresh
```

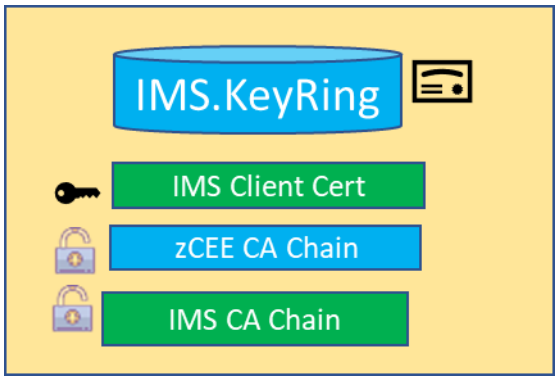
These commands

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

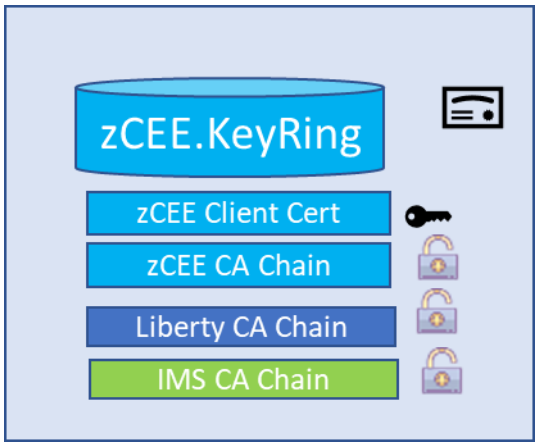


Mitch Johnson (mitchj@us.ibm.com)

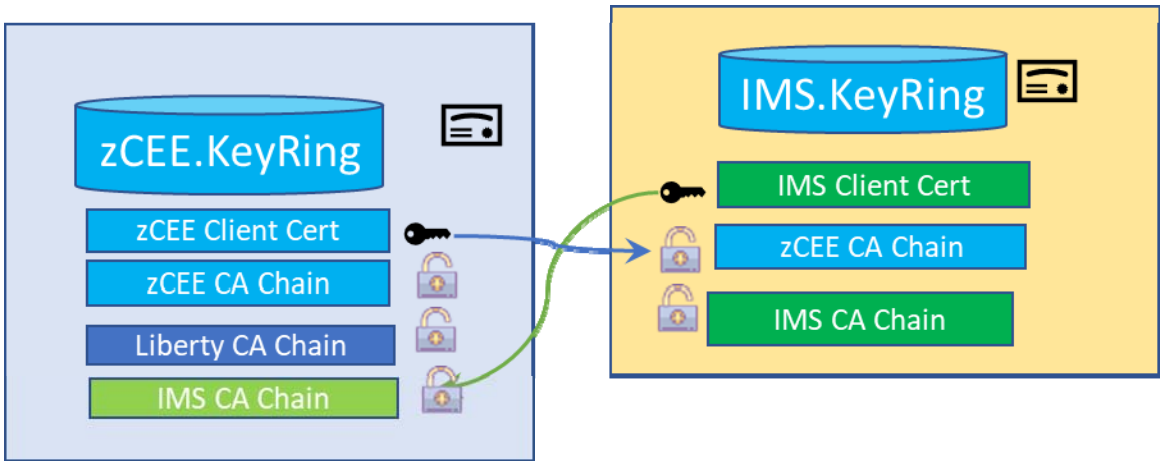
Below is visual representation of the key ring just created




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, found in directory */var/zosconnect/servers/myServer* 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/imsDatabasePassTicket.xml"/>
<include location="${server.config.dir}/includes/db2.xml"/>
<include location="${server.config.dir}/includes/shared.xml"/>
```

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

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

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

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

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

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



Mitch Johnson (mitchj@us.ibm.com)

___2. Enter MVS commands ***P BAQSRT and S BASSTRT*** 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 BAQSTRT,REFRESH,KEYSTORE***

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

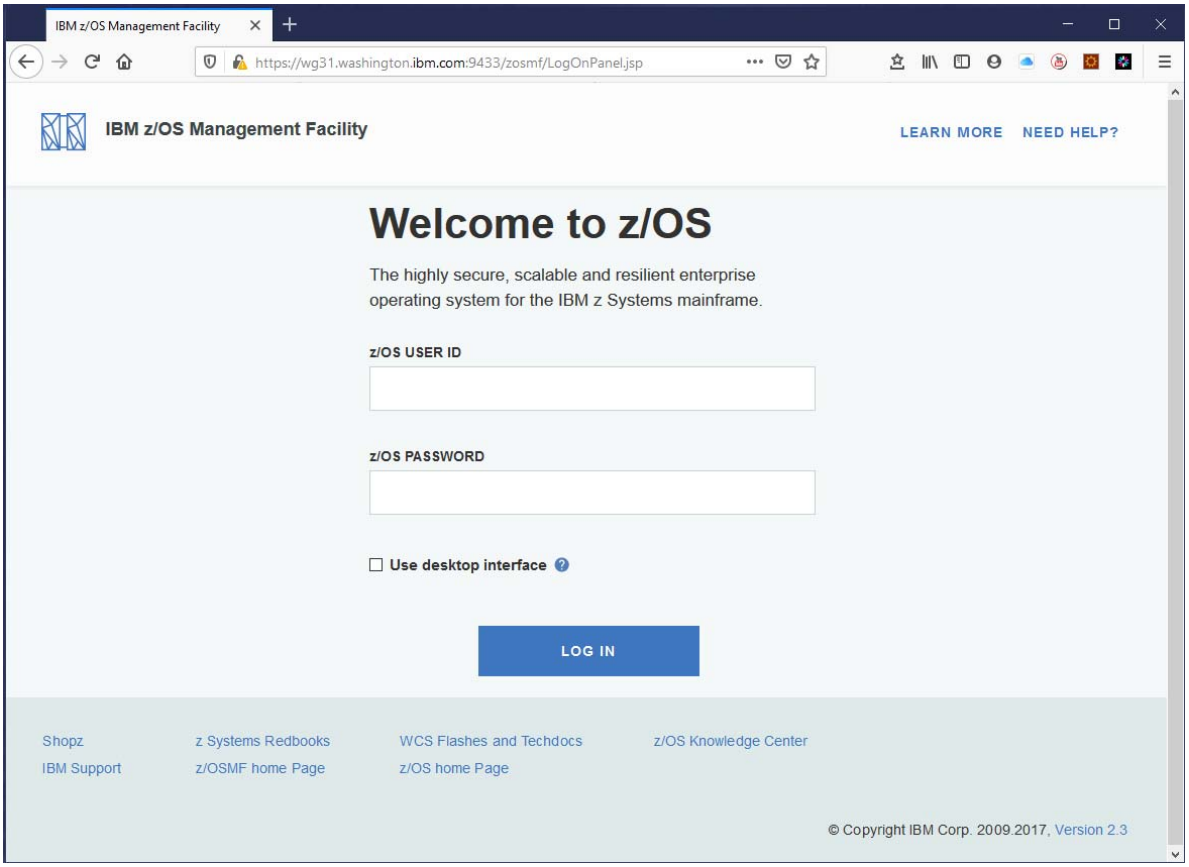


Mitch Johnson (mitchj@us.ibm.com)

## *Configure the AT-TLS policies*

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

___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.



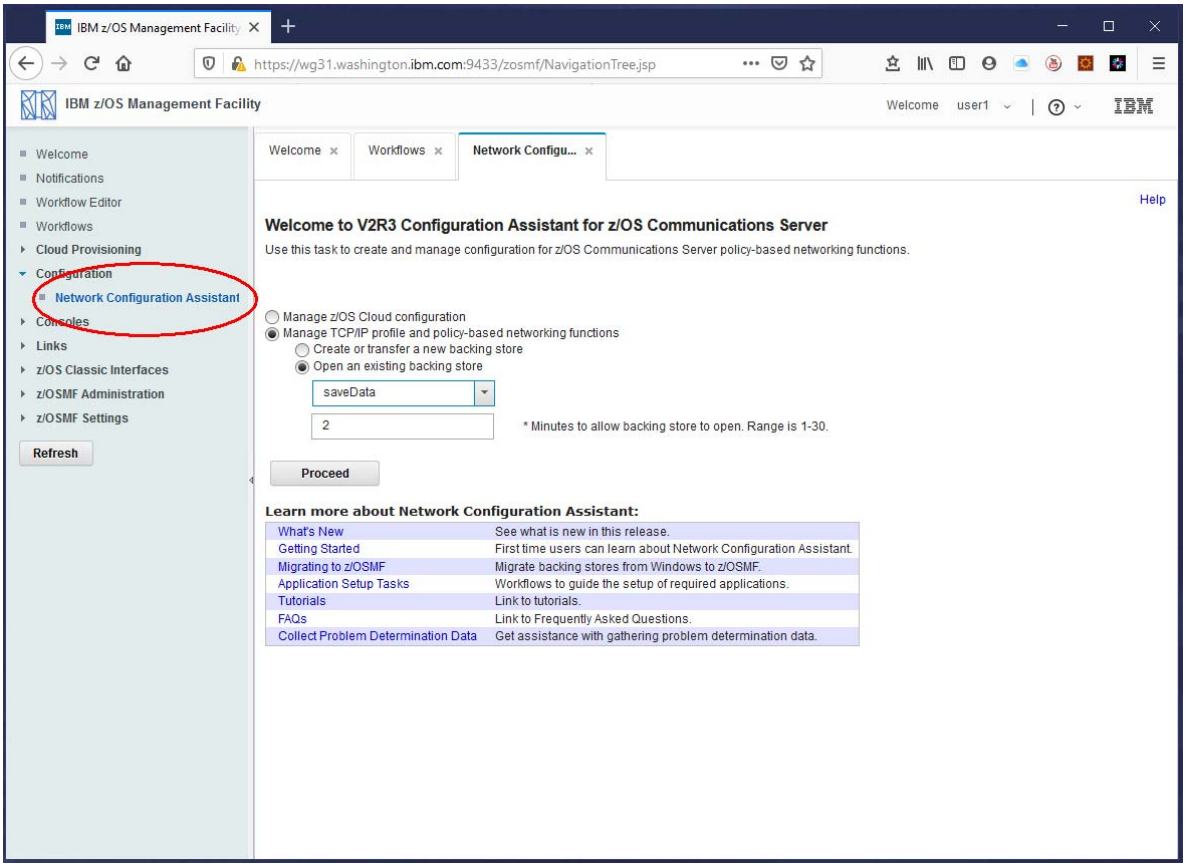

Note that some of the AT-TLS configuration steps described here may have been performed in another exercise.

___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. 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.



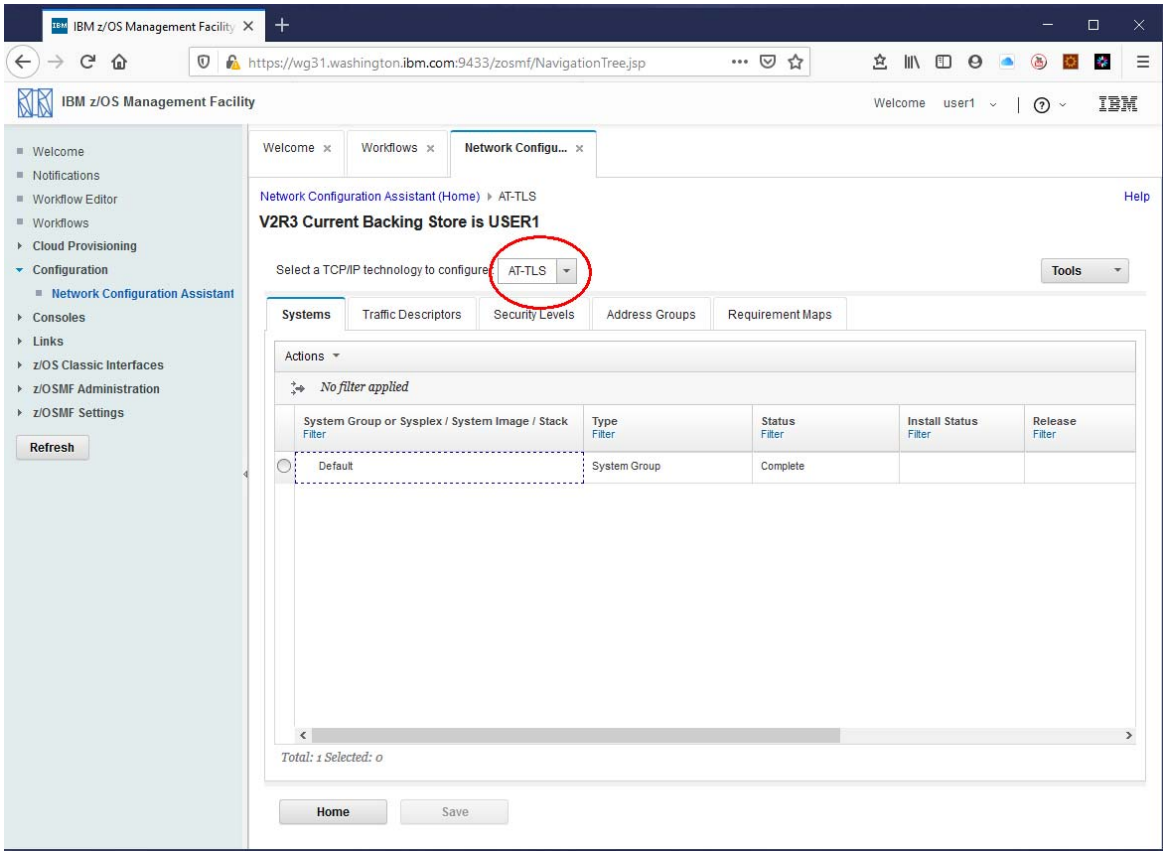

___4. Select the radio button beside *Create or transfer a new backing store* option and click the **Proceed** button.

___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.



Mitch Johnson (mitchj@us.ibm.com)

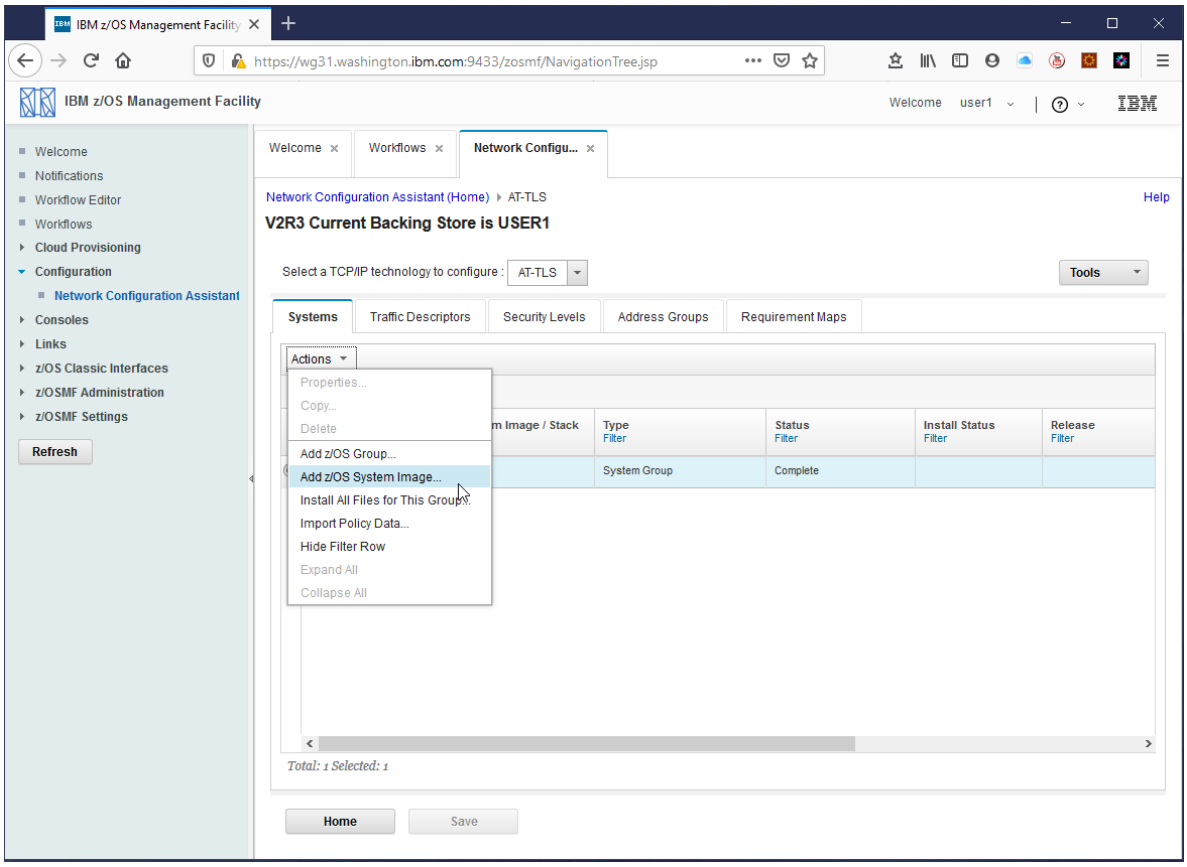
___6. On the *Network Configuration* tab use the pull-down arrow to select *AT-TLS* as the *TCP/IP technology to configure*.






Mitch Johnson (mitchj@us.ibm.com)

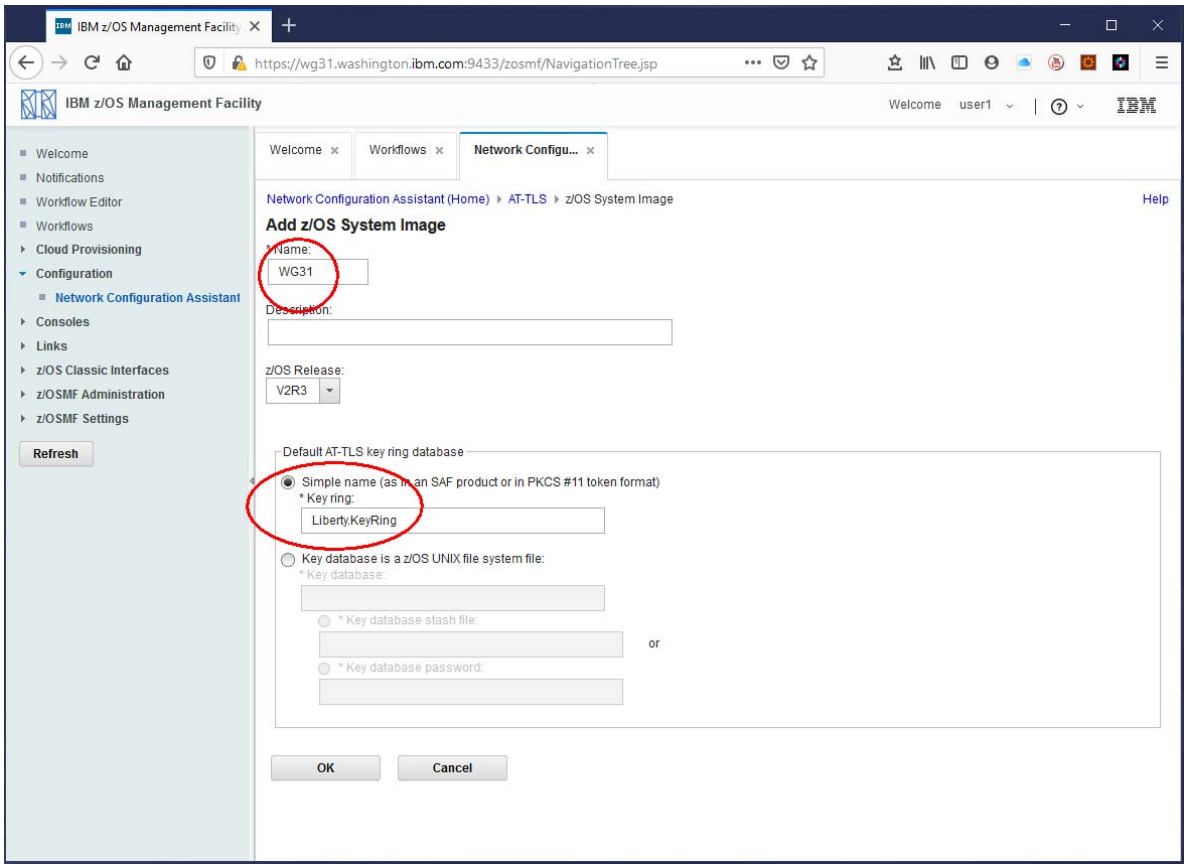
___7. Select the radio button beside the *Default - System Group* and use the *Action* pull-down button to select *Add z/OS System Image* option.






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




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

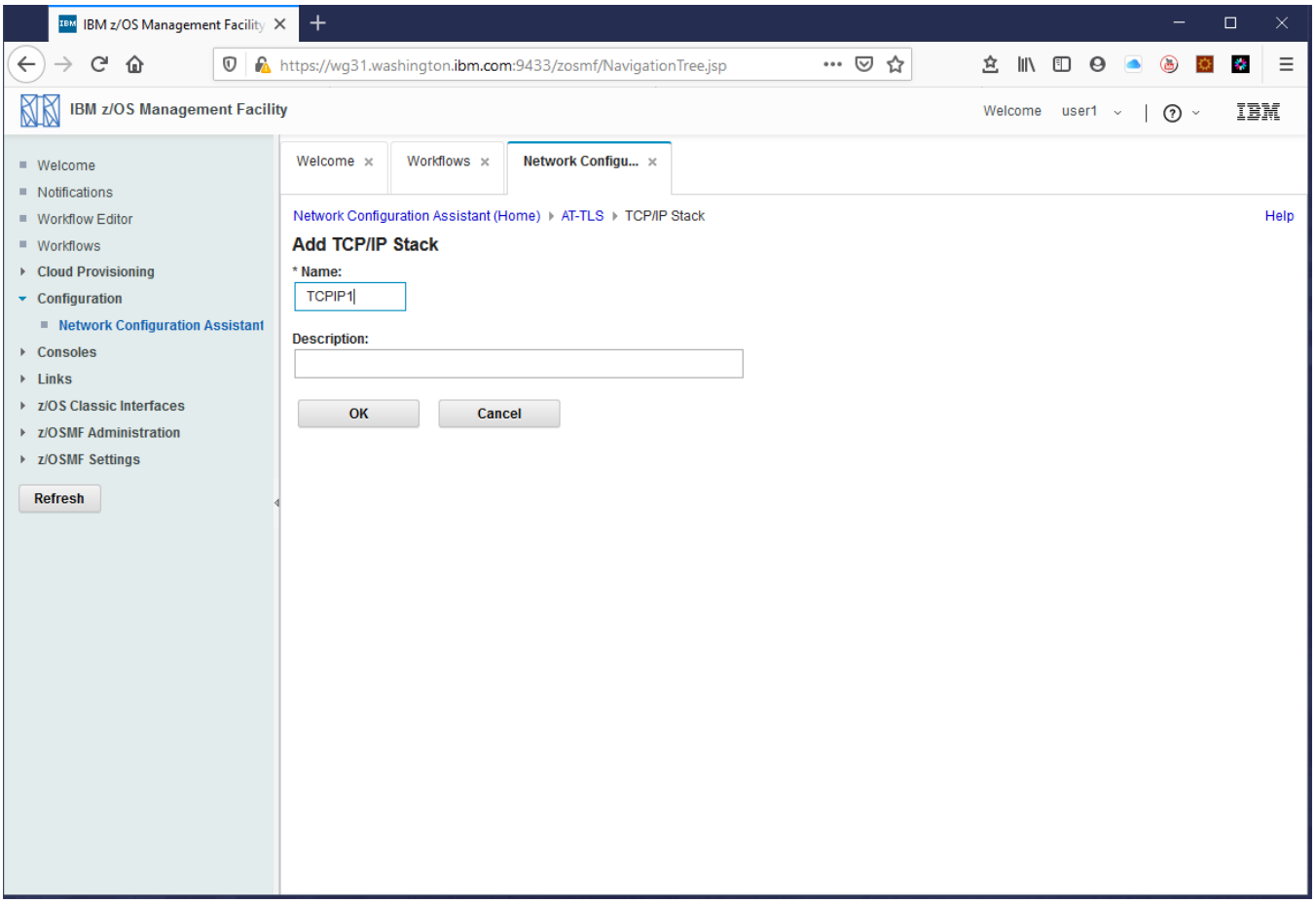
We recommend establishing a naming convention for key rings with each SAF identity using the same key ring name in the same context. Using this name as an example you could create a unique key ring named *Liberty.KeyRing* for SAF identities USER1, USER2, FRED, etc. Each user's key ring would have the same name but a different set of connected certificates. One default key ring specified at the image level covers all users.

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



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 ***TCPIP1*** as the name of the stack. Click **OK** to continue.

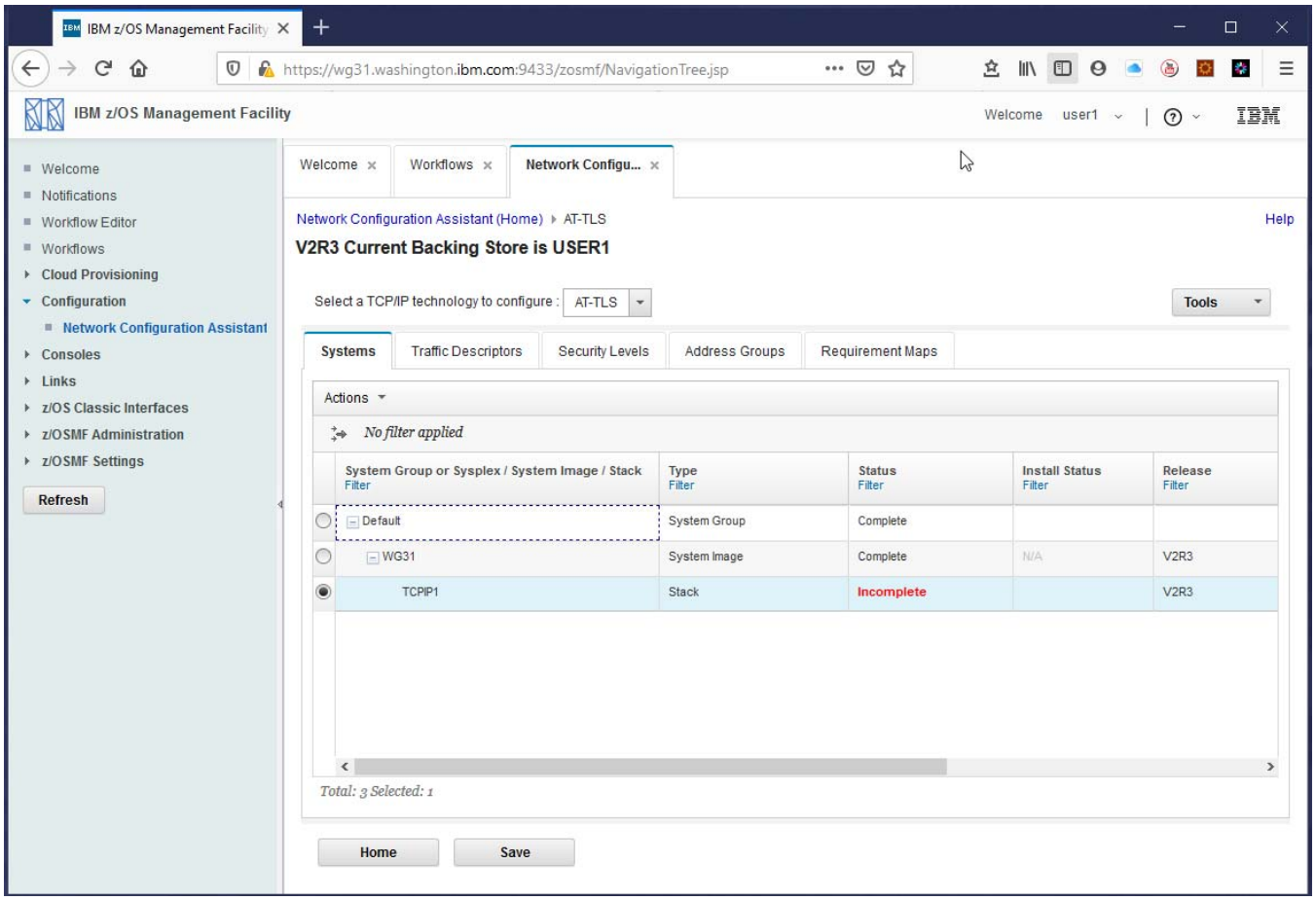



**Tech-Tip:** The value for the stack name was determine by the TCPIP Name display by entering the MVS command D TCPIP.

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

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


___12. This will display the window below:

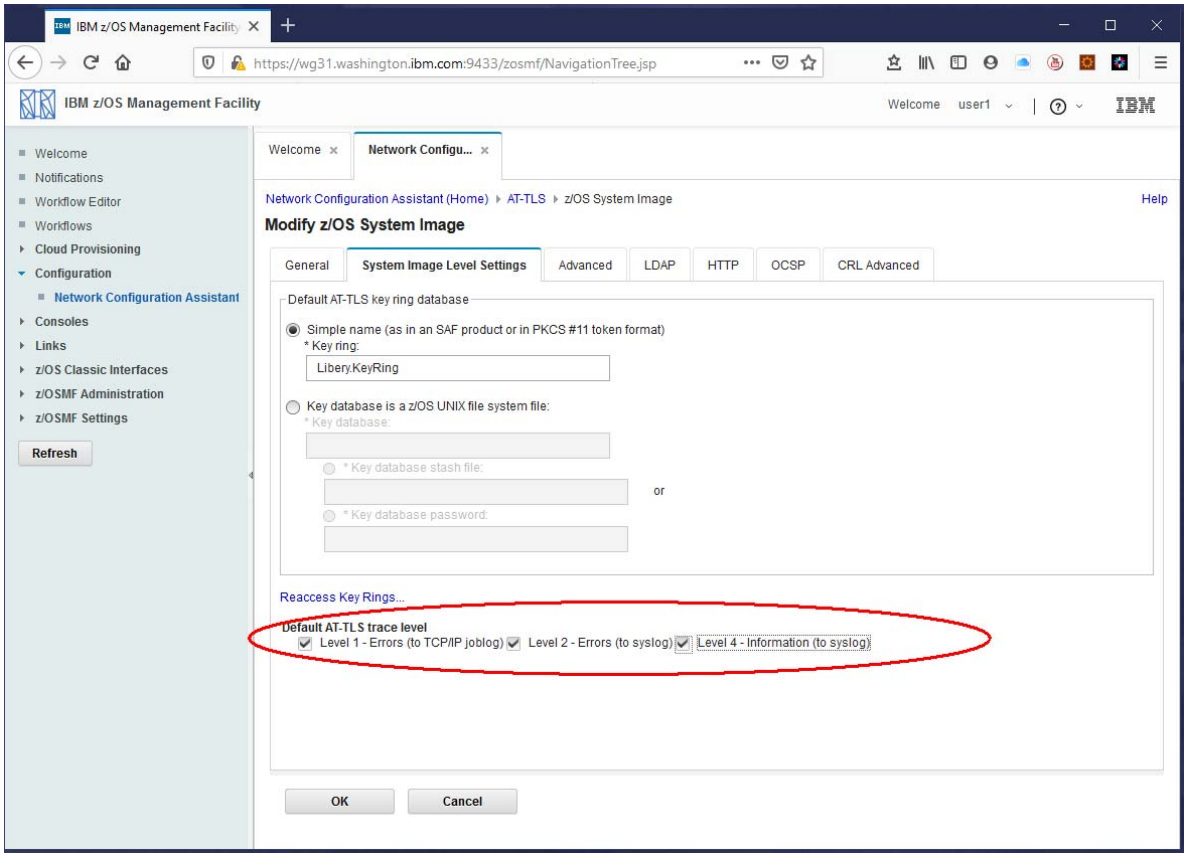



**Tech Tip:** The Incomplete warning will be addressed shortly.



Mitch Johnson (mitchj@us.ibm.com)

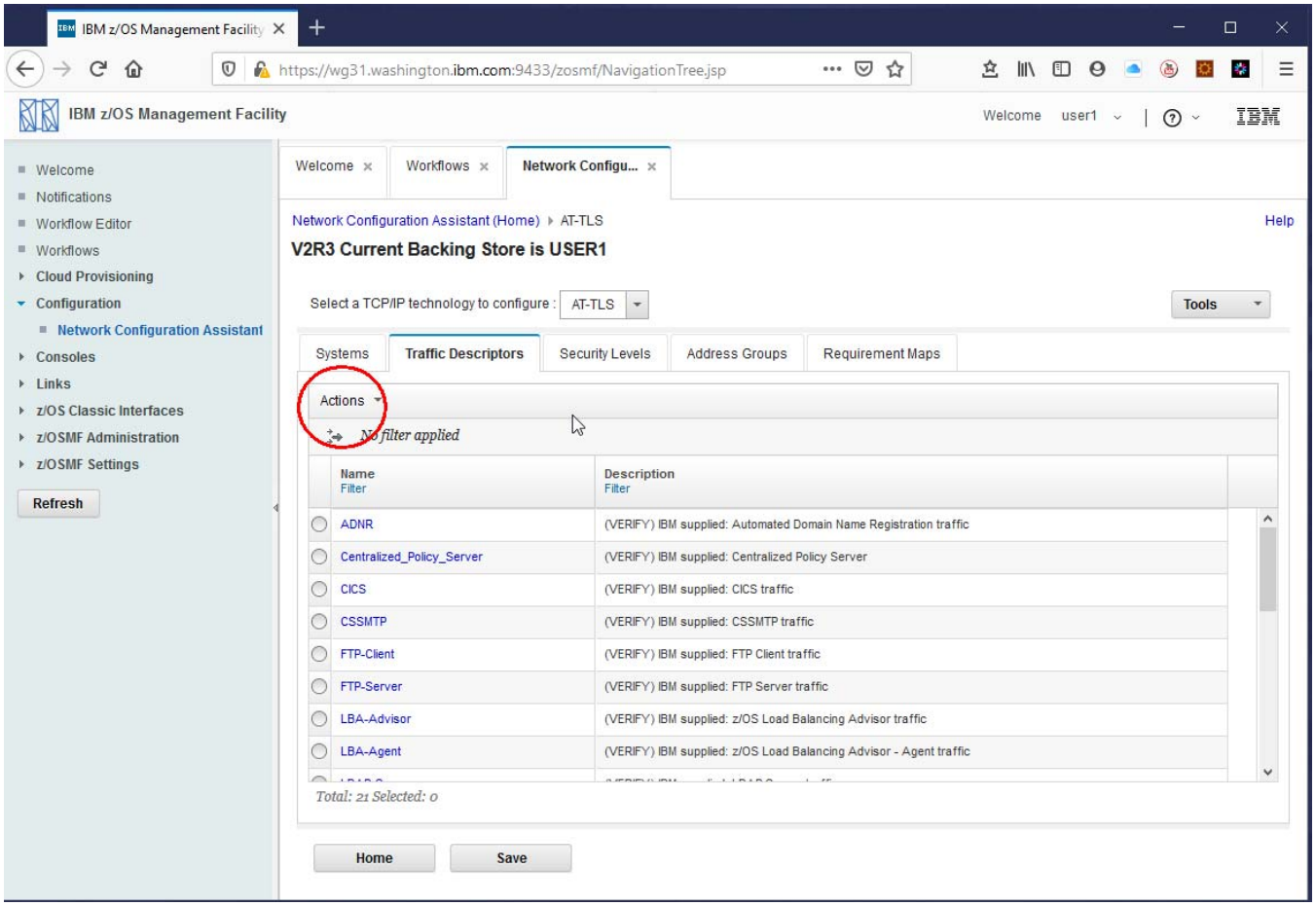
___13. Select the radio button beside *WG31* and use the *Actions* pull-down to select *Properties*. On the *Modify z/OS System Image* window select the *System Image Level Settings* tab and check all the trace level boxes as shown below. This is being done so we can confirm AT-TLS is being invoked and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






Mitch Johnson (mitchj@us.ibm.com)

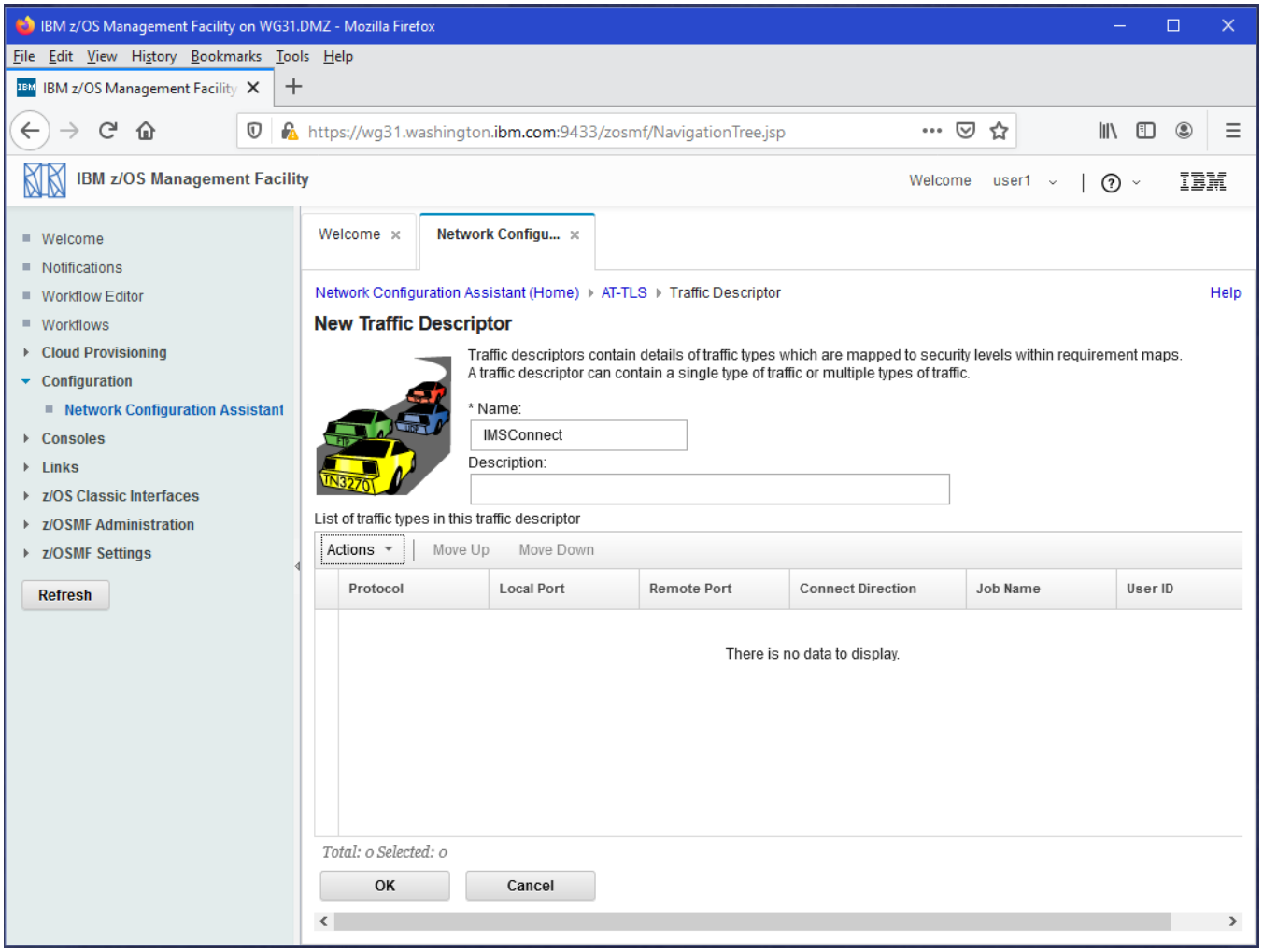
___14. Select the *Traffic Descriptors* tab and use the *Actions* pull-down to select *New*.






Mitch Johnson (mitchj@us.ibm.com)

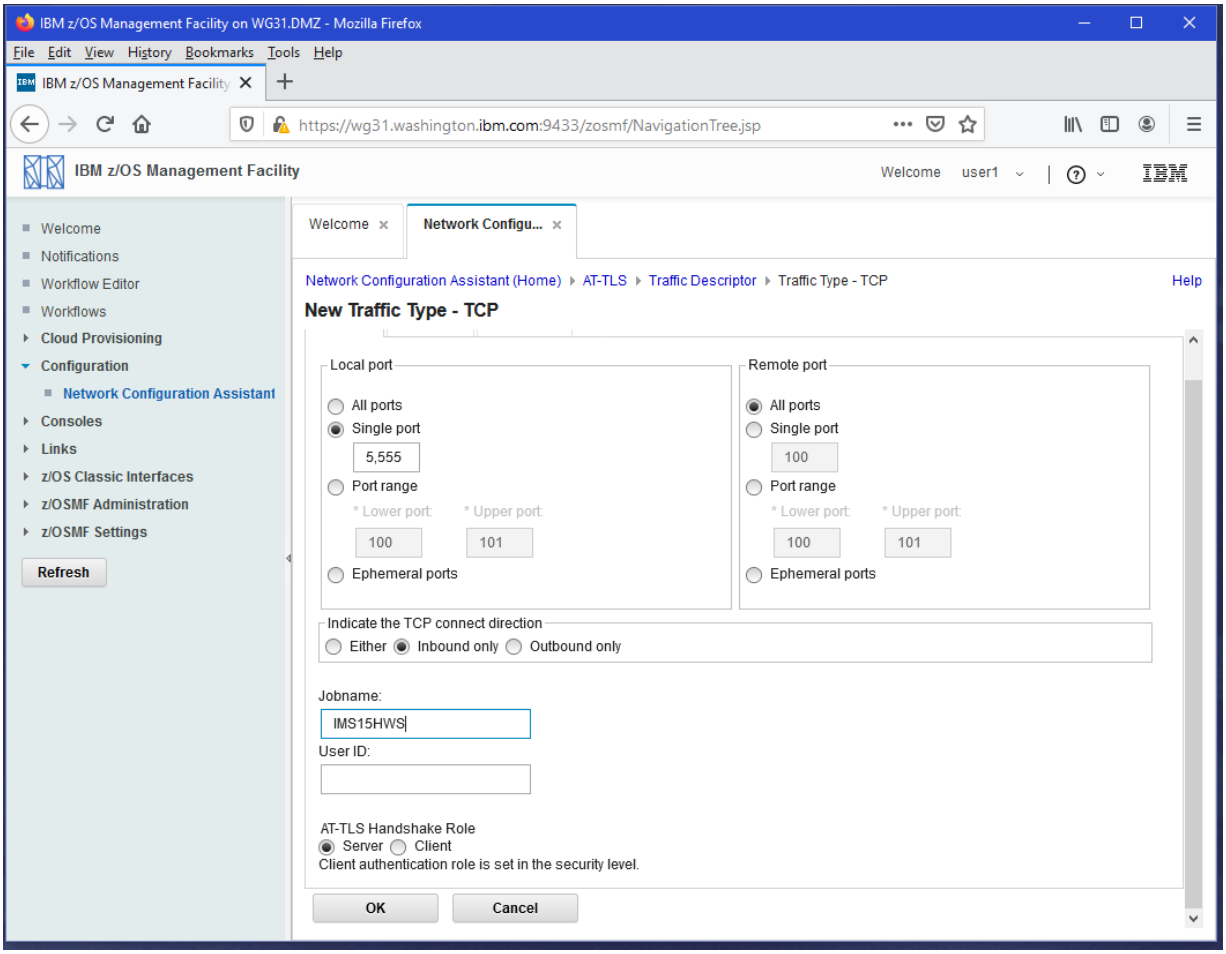
___15. On the *New Traffic Descriptor* window enter ***IMSConnect*** as the name and use the *Actions* pull-down and select *New* to start the definition of a new traffic descriptor type.






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 ***5555*** 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 ***IMS15HWS*** 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 *5555*. Port 5555 was identified in the IMS Connect configuration.

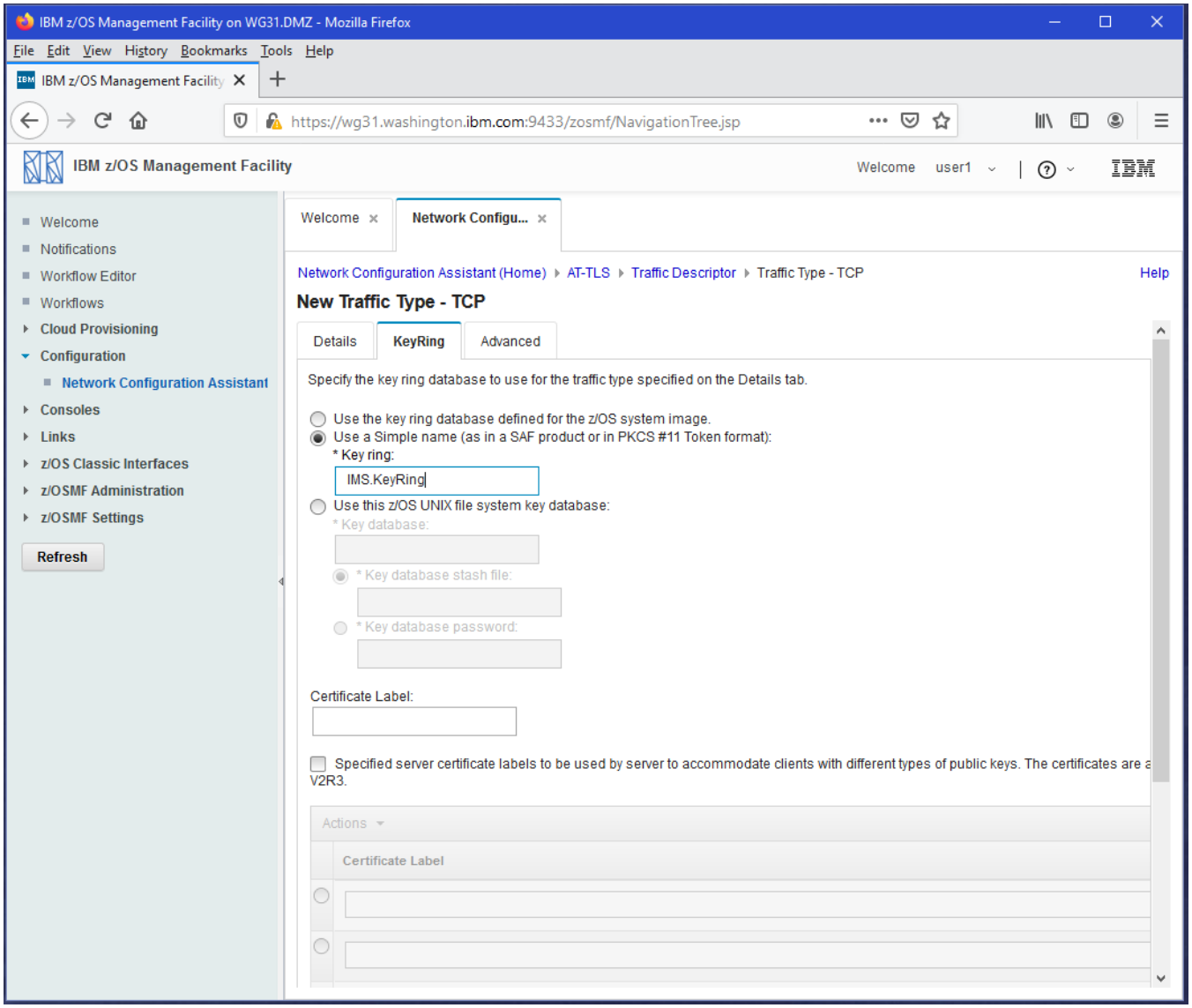
HWS=(ID=IMS15HWS,XIBAREA=100,RACF=Y,RRS=Y)
TCPIP=(HOSTNAME=TCPIP,PORTID=(4000,LOCAL),RACFID=JOHNSON,TIMEOUT=5000)
DATASTORE=(GROUP=OTMAGRP,ID=IVP1,MEMBER=HWSMEM,DRU=HWSYDRU0,
TMEMBER=OTMAMEM,APPL=IMSAPPL)
ODACCESS=(ODBMAUTOCONN=Y,IMSPLEX=(MEMBER=IMS15HWS,TMEMBER=PLEX1),
DRDAPORT=(ID=**5555**,PORTTMOT=6000),ODBMTMOT=6000,APPL=IMSAPPL)

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. IMS.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)

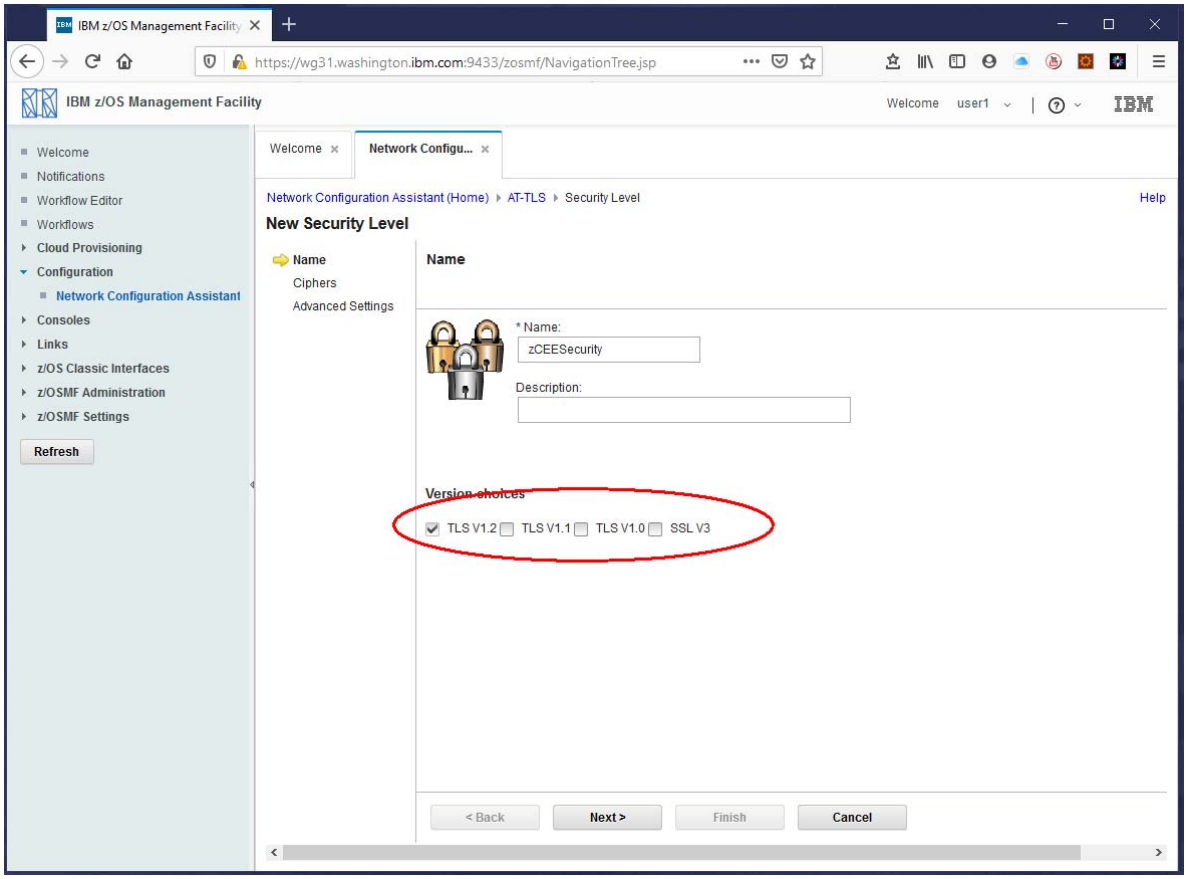
___17. On the *KeyRing* tab select the radio button beside *Use a Simple name* and enter ***IMS.KeyRing*** as the key ring name. Click **OK** twice to continue.






Mitch Johnson (mitchj@us.ibm.com)

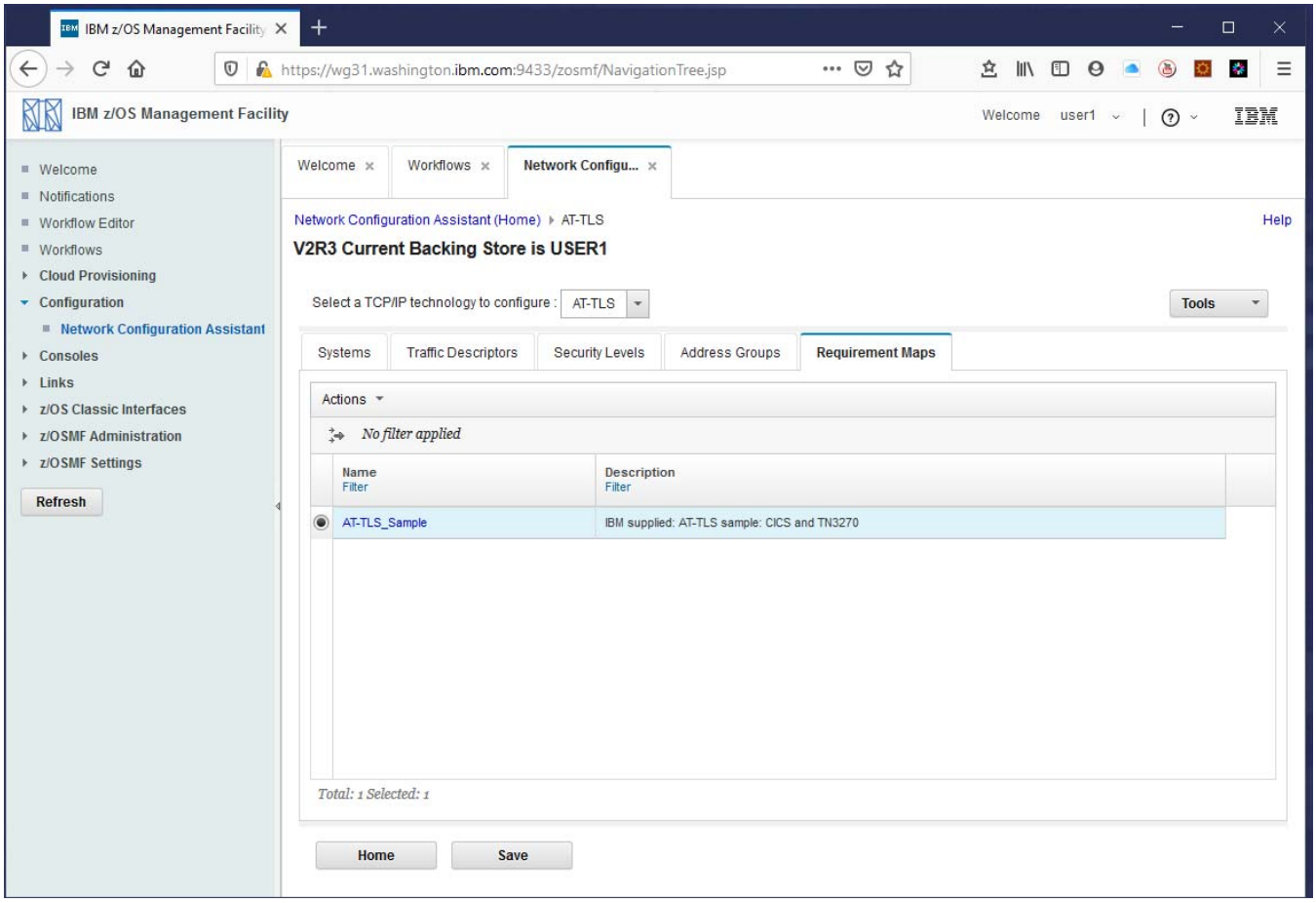
___18. Next, click the Security Levels tab and use the *Actions* pull-down button and to select the *New* option. On the *New Security Level* windows, enter ***zCEESecurity*** for the *Name* and check the box beside *TLS V1.2* and uncheck the other boxes. Click **Next** to display the *Cipher selection*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






Mitch Johnson (mitchj@us.ibm.com)

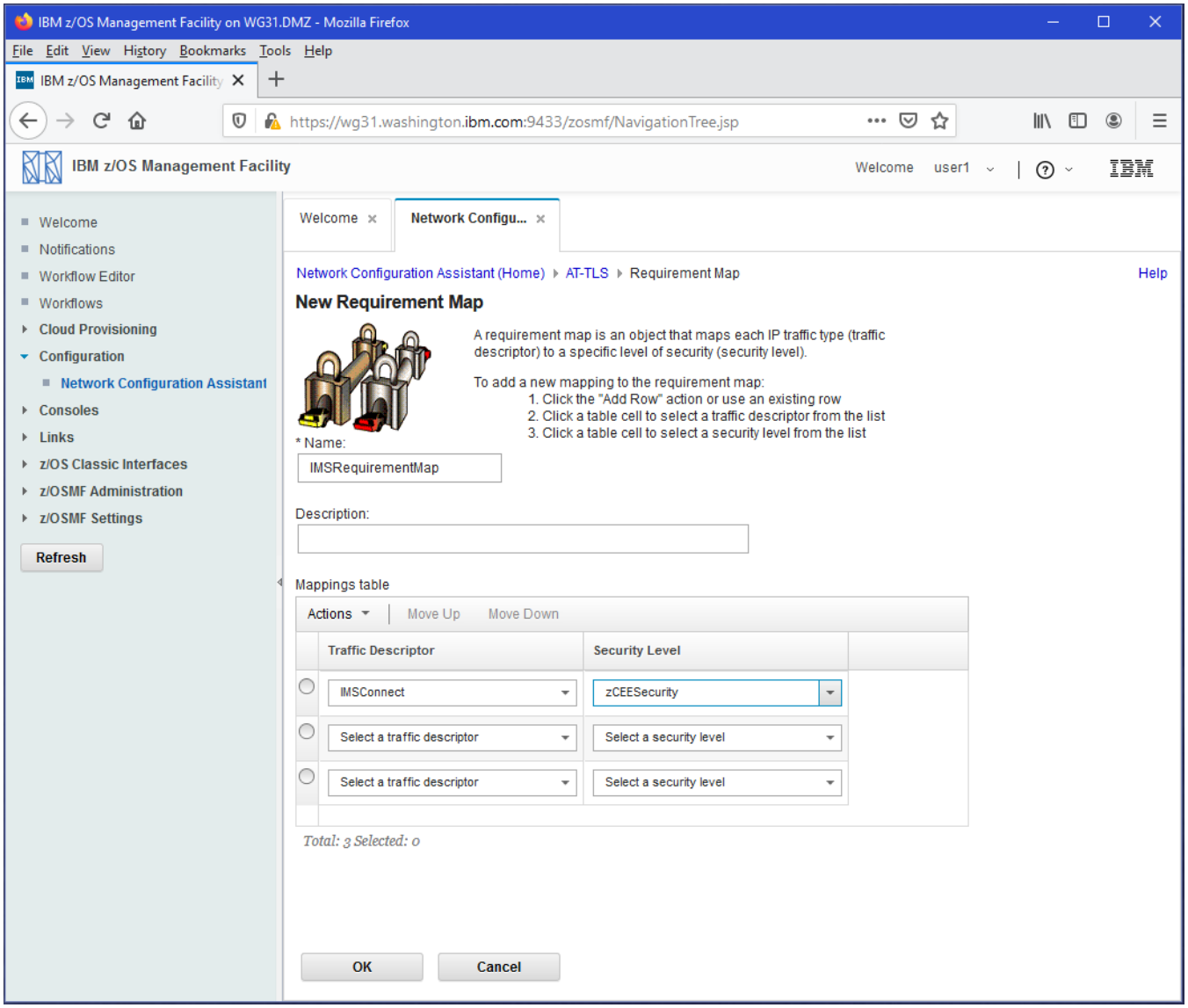
___19. Next, click the *Requirement Maps* tab. Use the *Actions* pull-down button and to select the *New* option.






Mitch Johnson (mitchj@us.ibm.com)

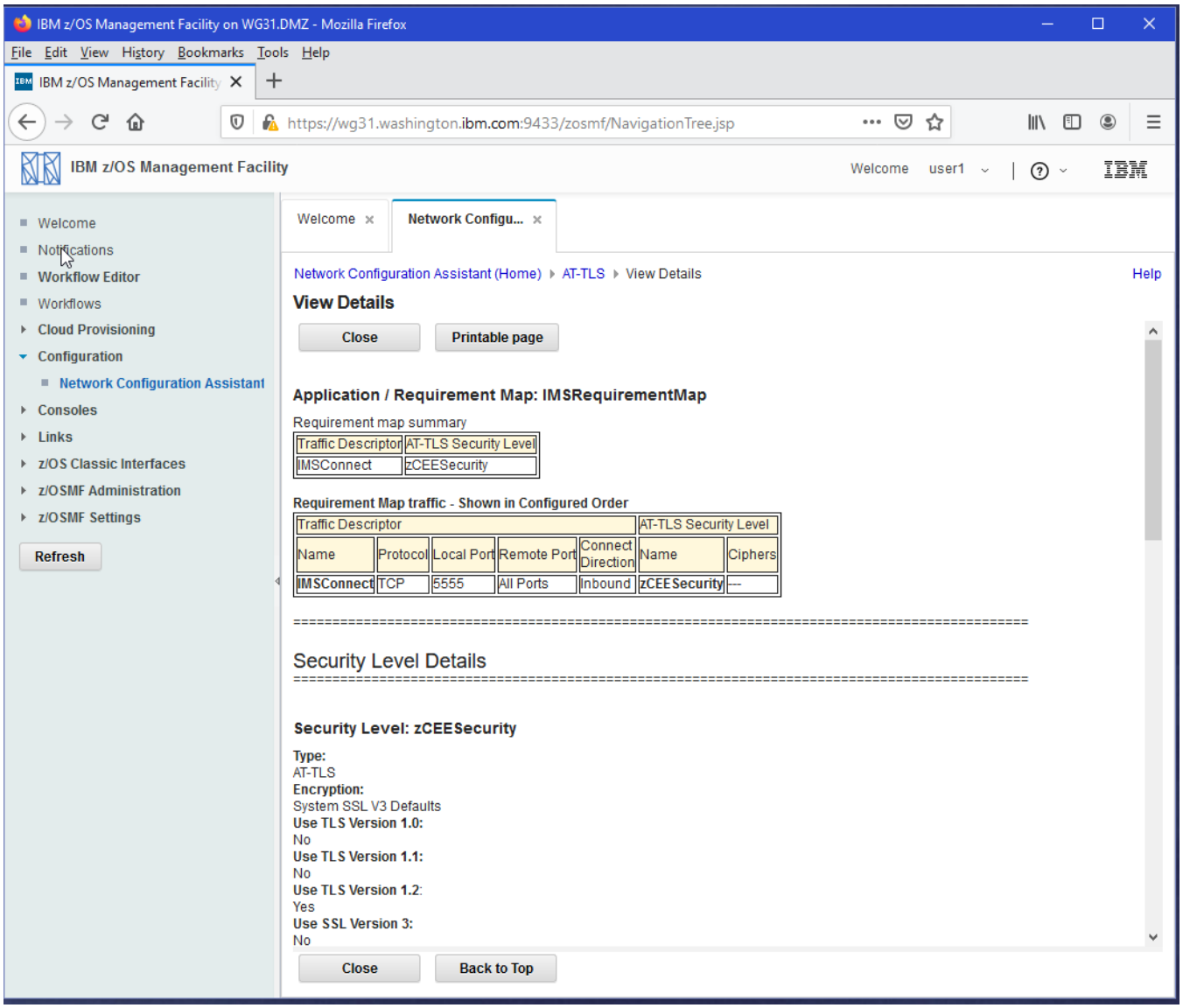
___20. On the *New Requirement Map* window enter ***IMSRequirementMap*** as the *Name* and use the pull-down arrows to select *IMSConnect* as the *Traffic Descriptor* and *zCEESecurity* as the *Security Level* for this map. Click **OK** to continue.






Mitch Johnson (mitchj@us.ibm.com)

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

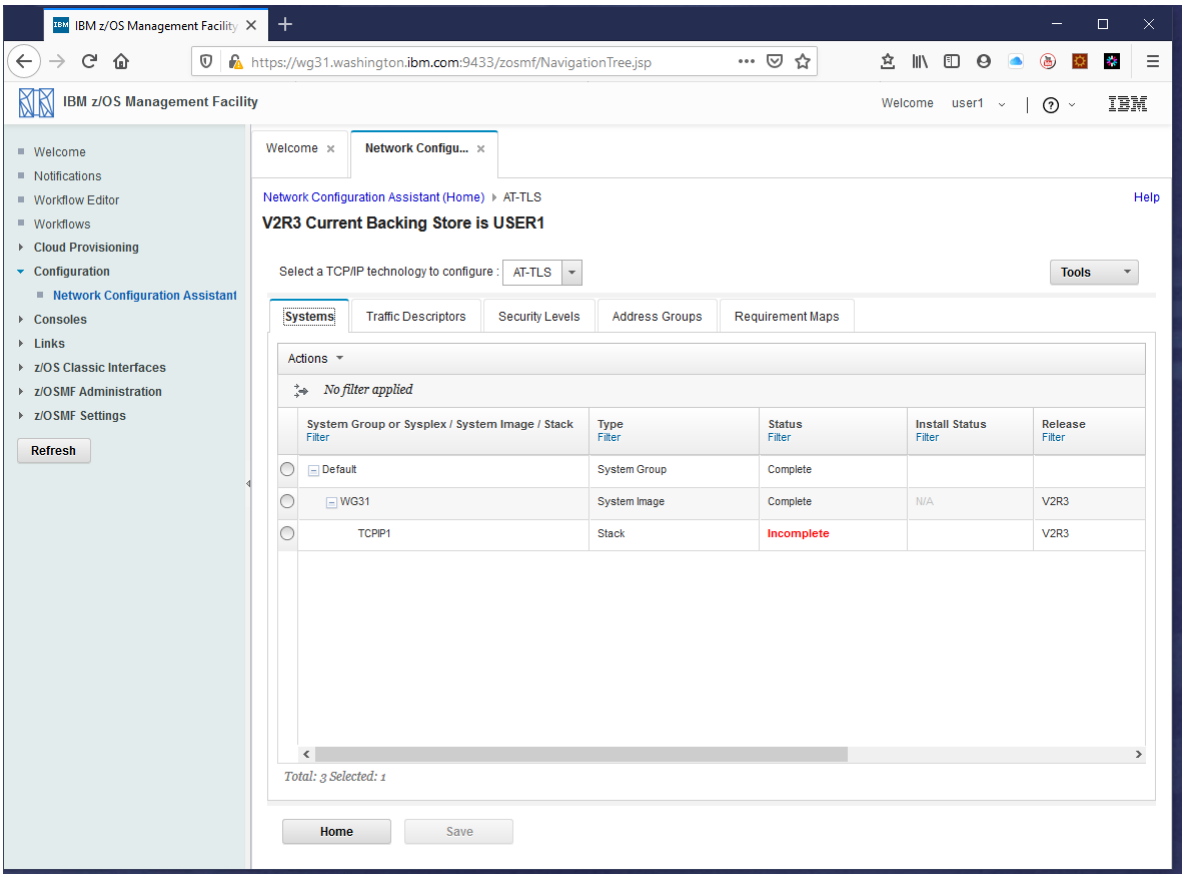



___22. Click the **Save** button to save the configuration.



Mitch Johnson (mitchj@us.ibm.com)

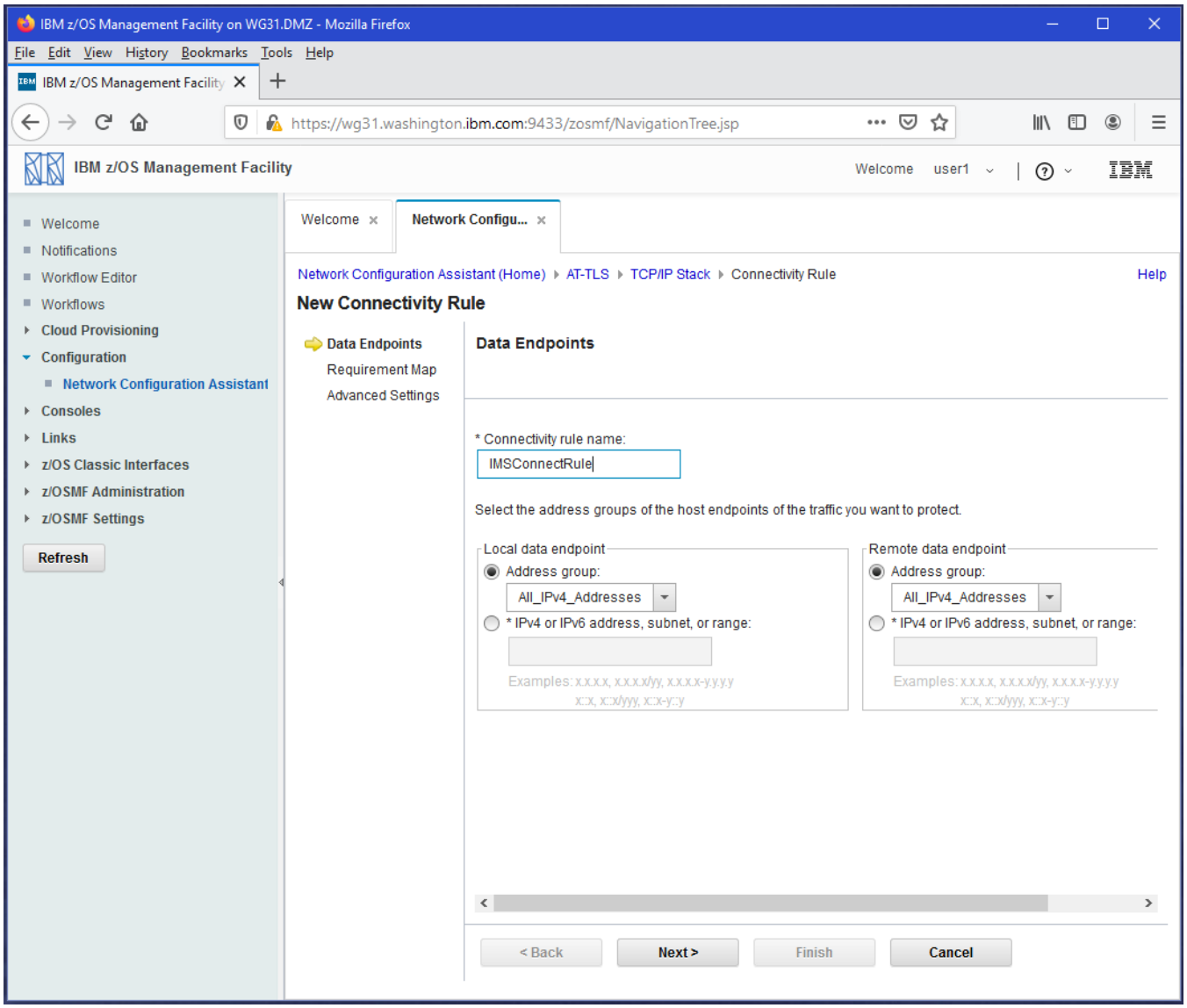
___23. When the save has complete click on the *Systems* tab to return to this window.






Mitch Johnson (mitchj@us.ibm.com)

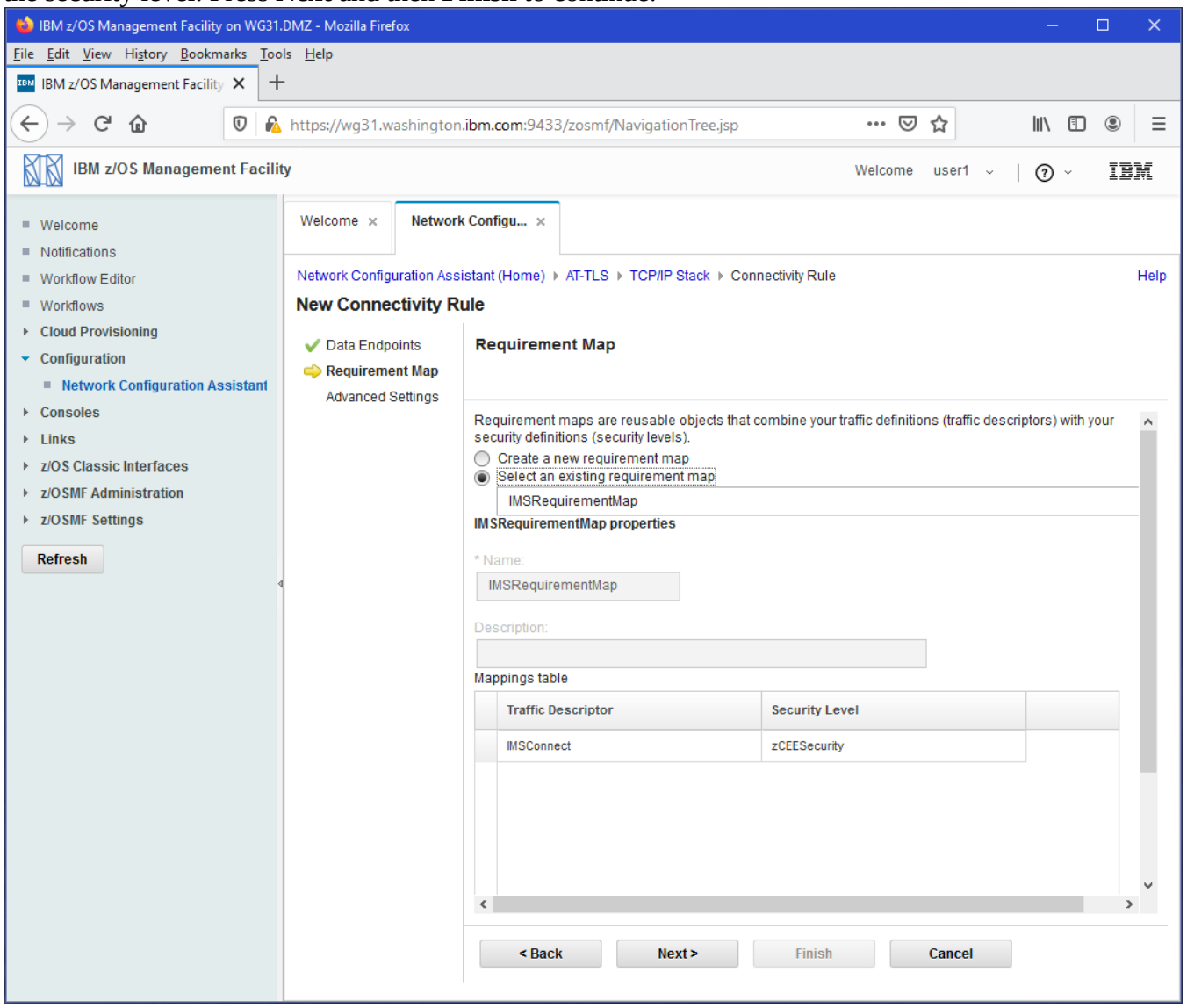
___24. Select the radio button beside *TCPIP1*. Use the *Actions* pull-down to select *Rules.* This is where these definitions will be tied together. Use the *Actions* pull-down again and select *New* to create a new connectivity rule. Enter ***IMSConnectRule*** for the *Connectivity rule name* and press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






Mitch Johnson (mitchj@us.ibm.com)

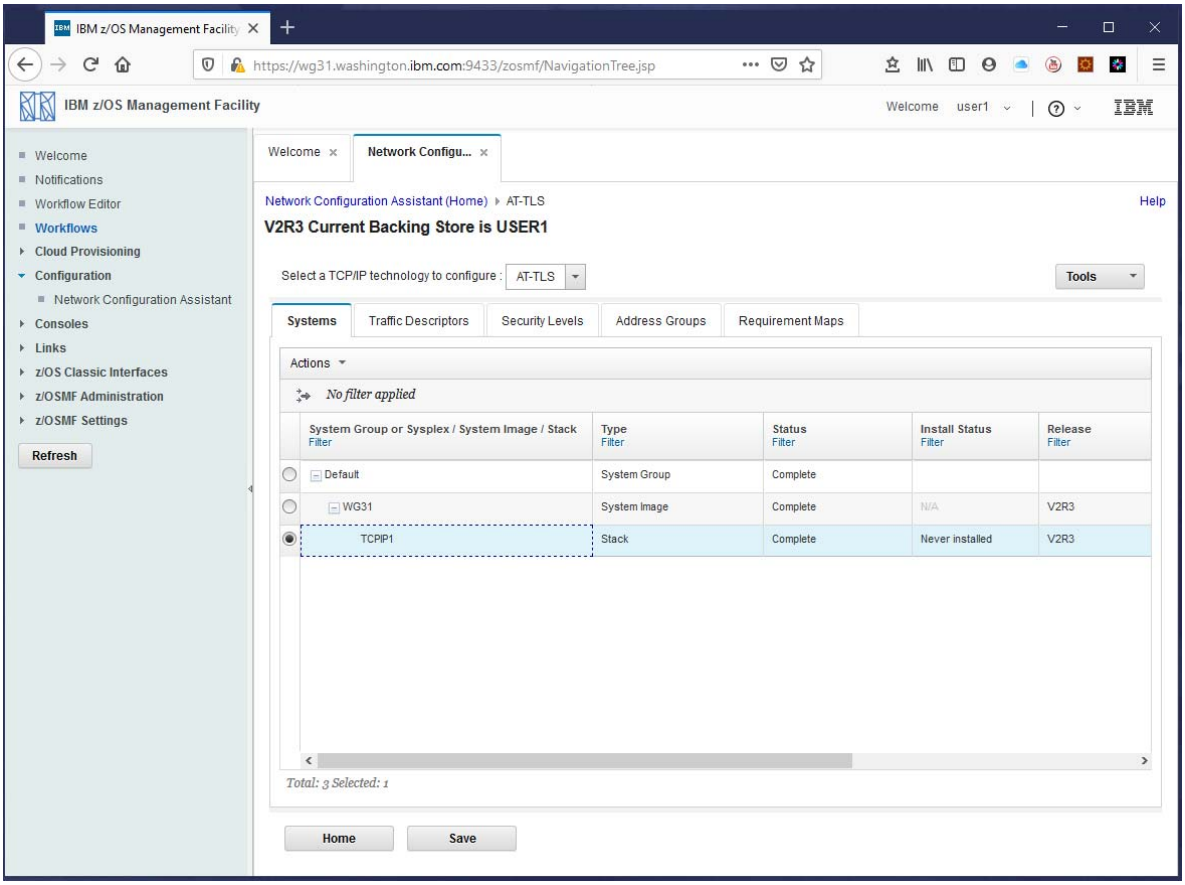
___25. On the *New Connectivity Rule – Requirement Map* window select the radio button beside *Select an existing requirement map* and use the pull-down to select *IMSRequirementMap*. This should automatically populate the mapping table with *IMSConnect* as the traffic descriptor and *zCEESecurity* as 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.

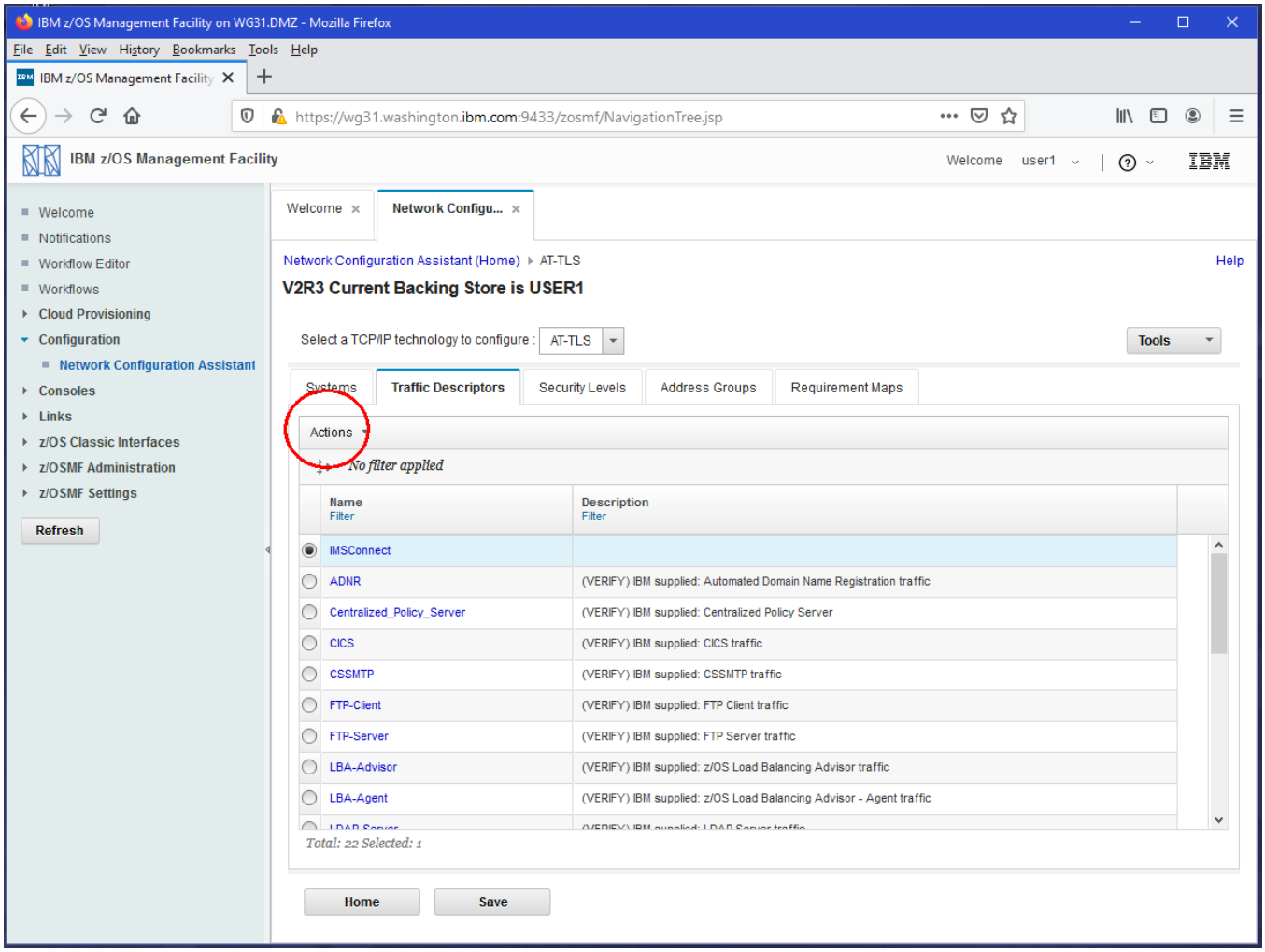



This completes the configuration of the inbound policy for the server side of the handshake. Now an outbound policy for the client side of the handshake needs to be configured



Mitch Johnson (mitchj@us.ibm.com)

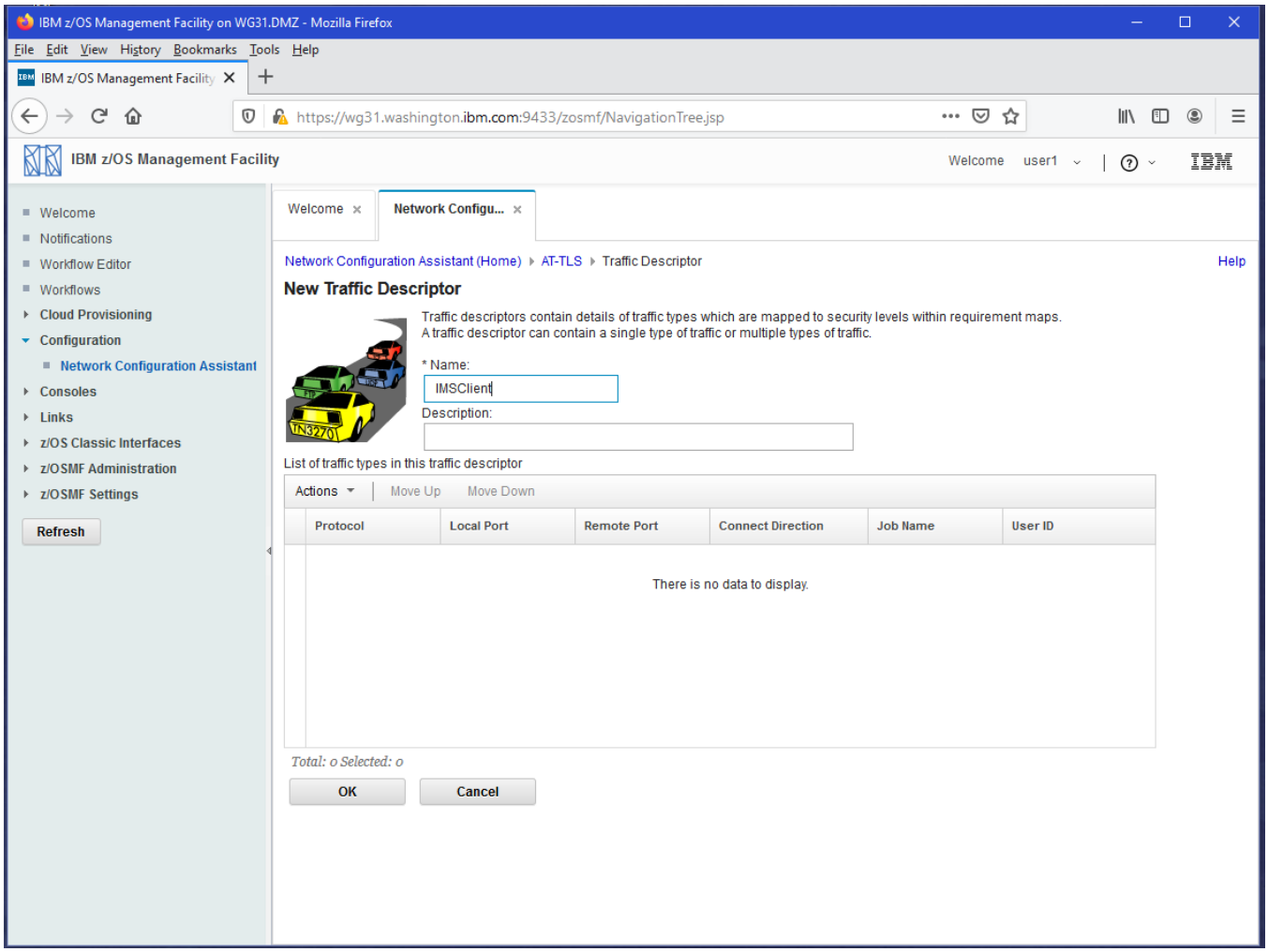
___27. Select the *Traffic Descriptors* tab. Use the *Actions* pull-down to select *New*.






Mitch Johnson (mitchj@us.ibm.com)

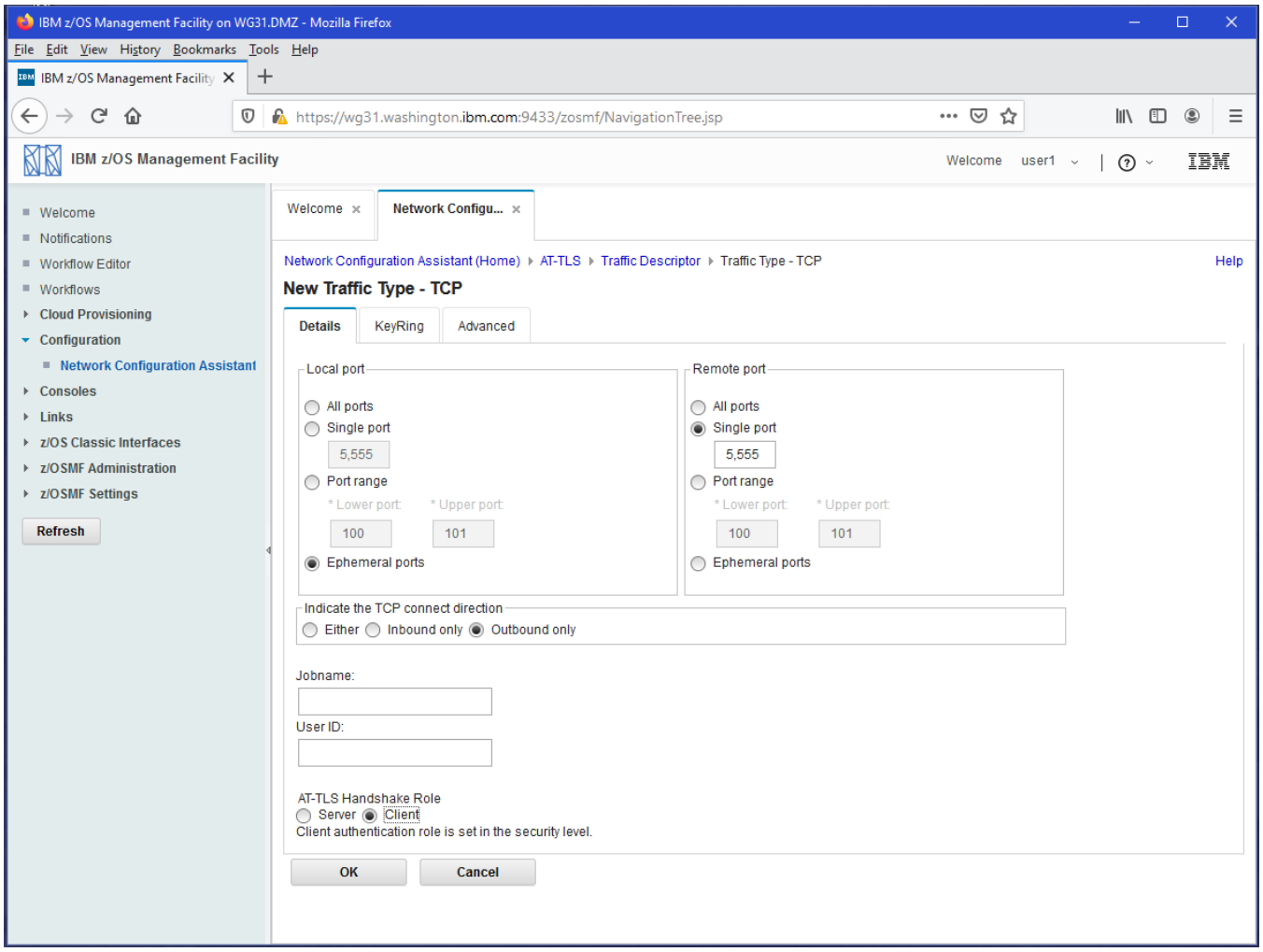
___28. On the *New Traffic Descriptor* window enter ***IMSClient*** as the name. Use the *Actions* pull-down and select *New* to start the definition of a new traffic descriptor type.






Mitch Johnson (mitchj@us.ibm.com)

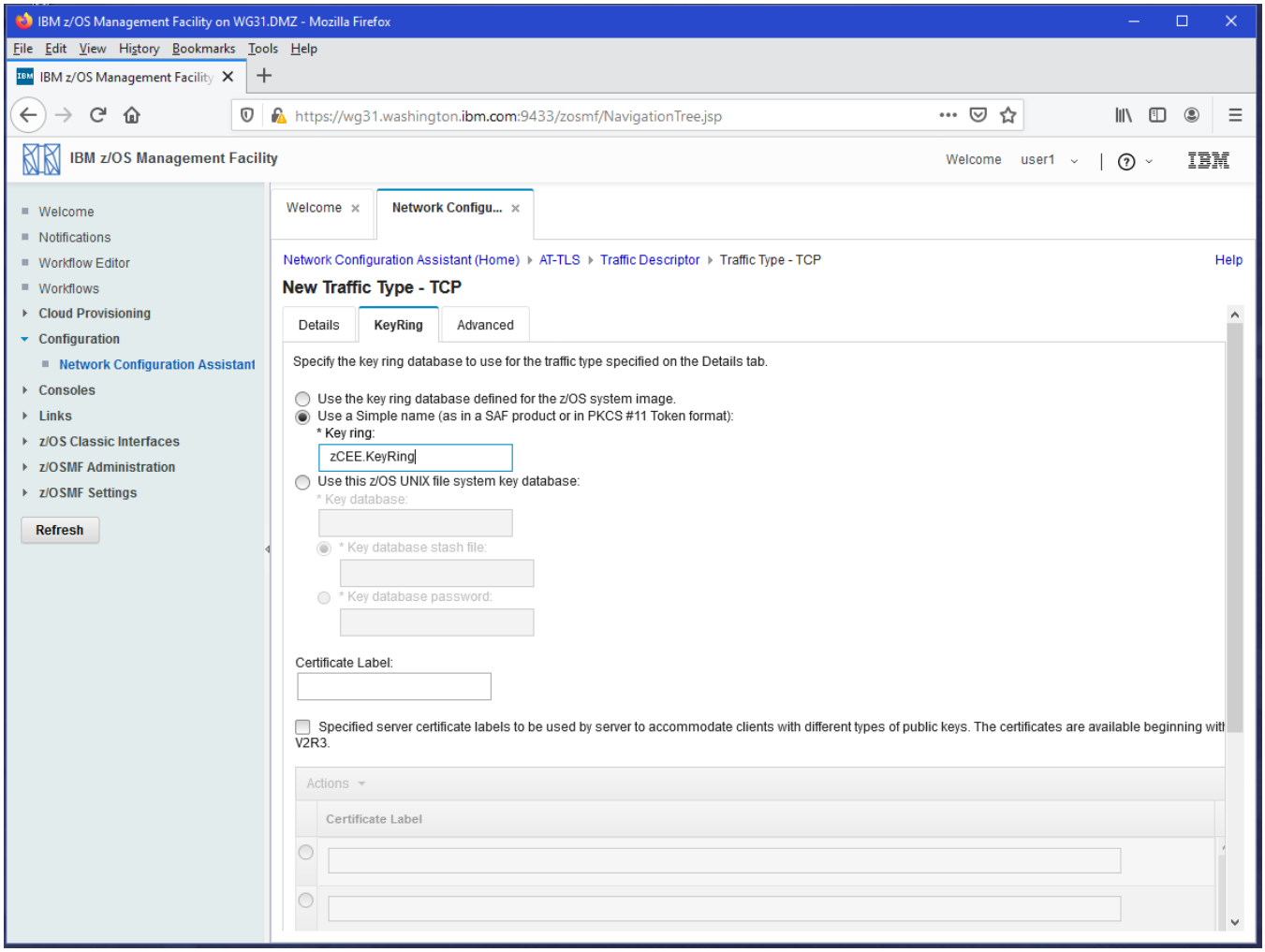
___29. On the *New Traffic Type – TCP* window, select the radio button beside *Ephemeral ports* under *Local port/*. Select the radio button *Single ports* under *Remote port* and enter ***5555***. Select the radio button beside *Outbound only* under *Indicate the TCP connection direction*. Finally select the radio button beside *Client* under *AT-TLS Handshake Role*. Next click on the *KeyRing* tab to continue.






Mitch Johnson (mitchj@us.ibm.com)

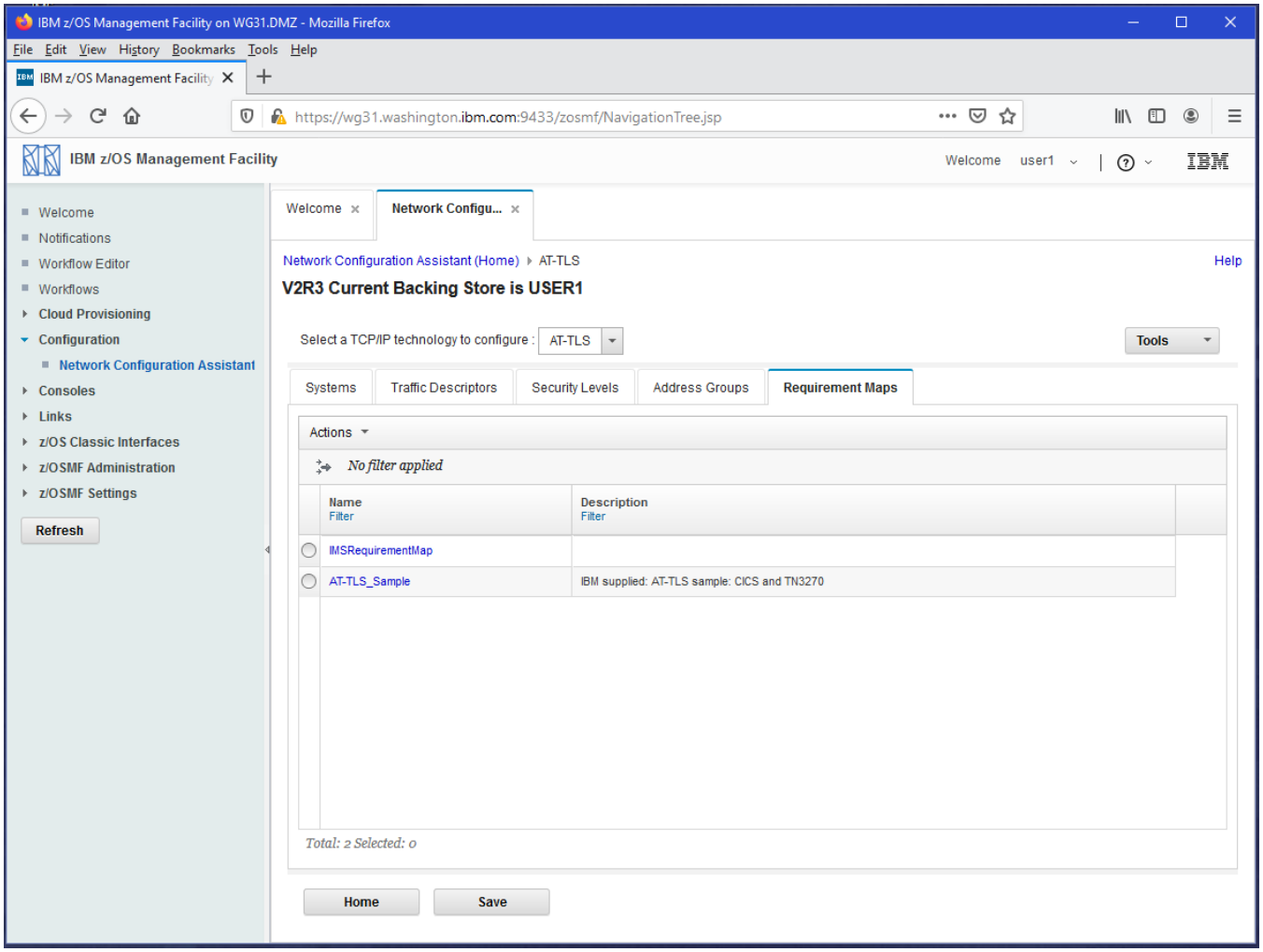
___30. On the *KeyRing* tab select the radio button beside *Use a Simple name* and enter ***zCEE.KeyRing*** as the key ring name. Click **OK** twice to continue.






Mitch Johnson (mitchj@us.ibm.com)

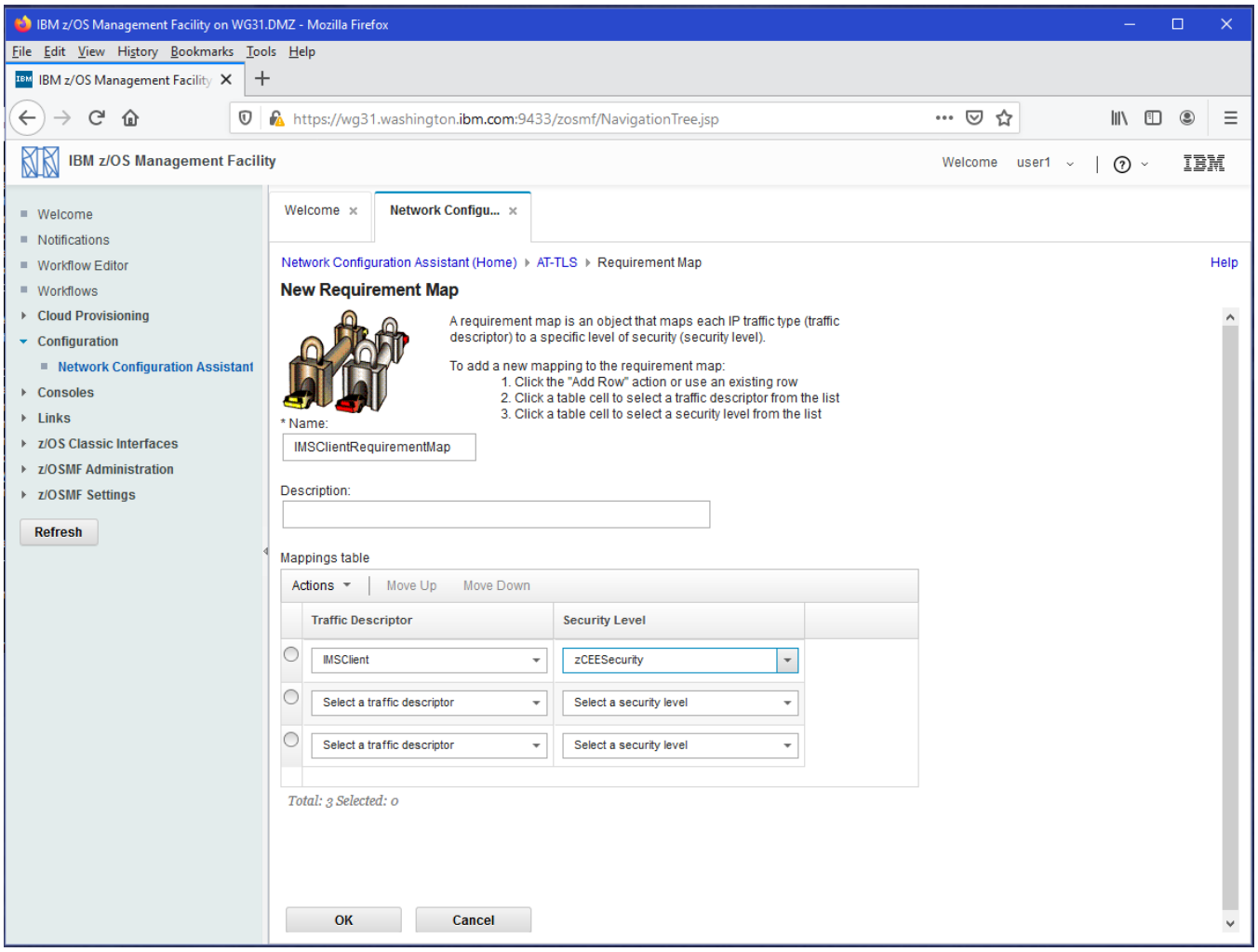
___31. Next, click the *Requirement Maps* tab. Use the *Actions* pull-down button and to select the *New* option.






Mitch Johnson (mitchj@us.ibm.com)

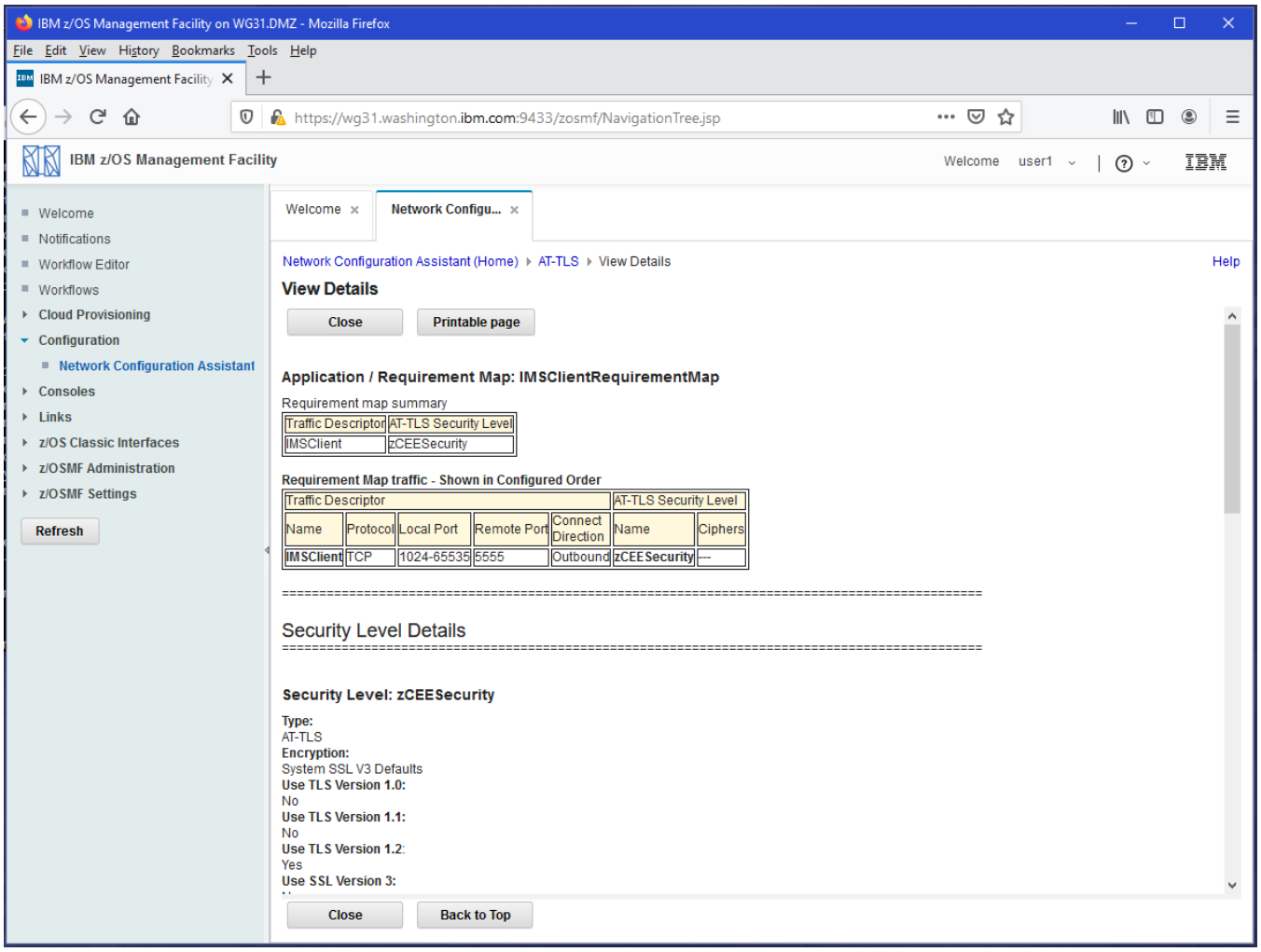
___32. On the *New Requirement Map* window enter ***IMSClientRequirementMap*** as the *Name*. Use the pull-down arrows to select *IMSClient* as the *Traffic Descriptor* and *zCEESecurity* as the *Security Level* for this map. Click **OK** to continue.






Mitch Johnson (mitchj@us.ibm.com)

___33. Select the radio button beside *IMSClientRequirementMap, U*se 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.




___34. Click the **Save** button to save the configuration.

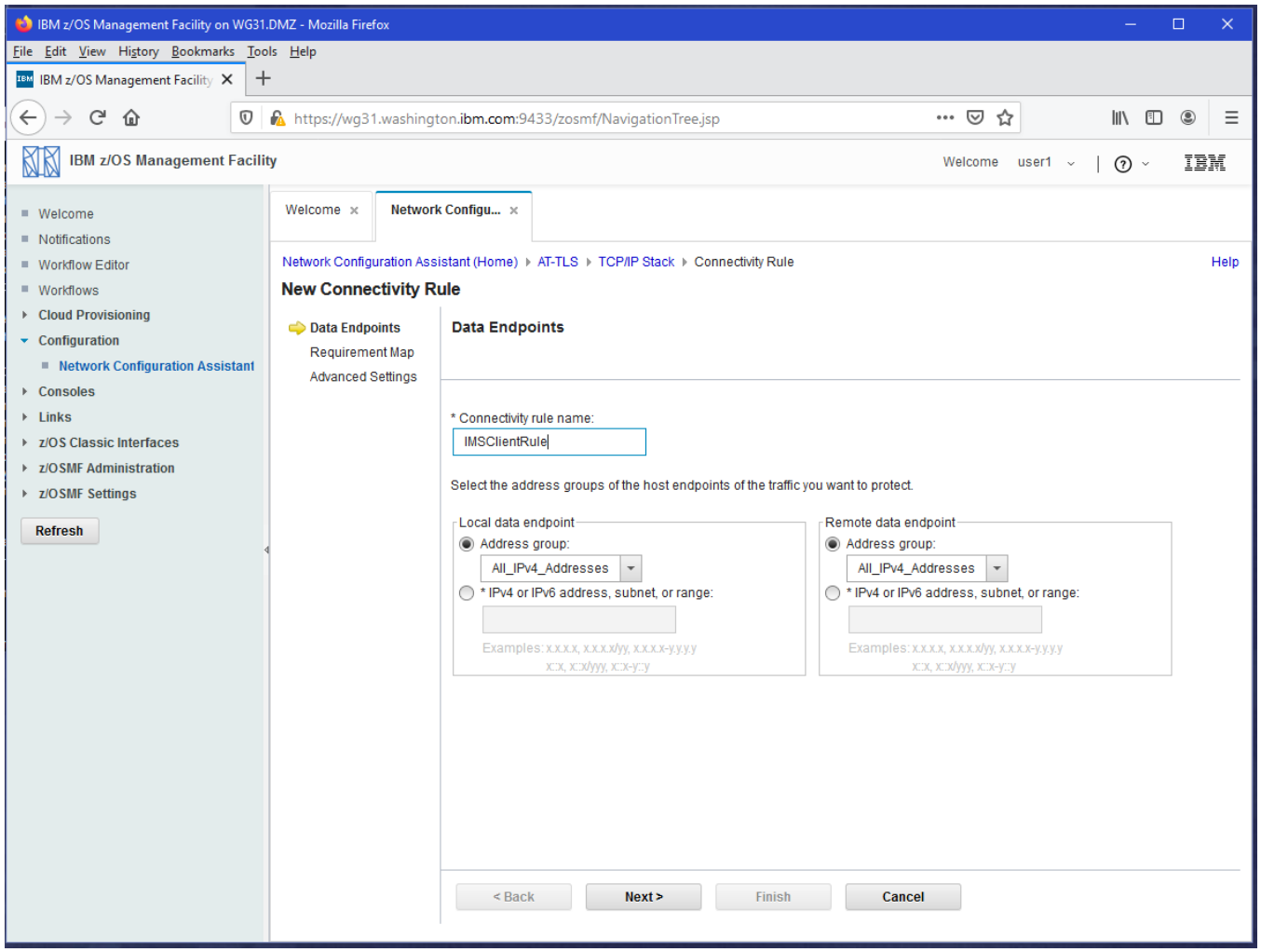


Mitch Johnson (mitchj@us.ibm.com)

___35. When the save has complete click on the *Systems* tab to return to this window.

A connectivity rule for the IMS Client now needs to be added.

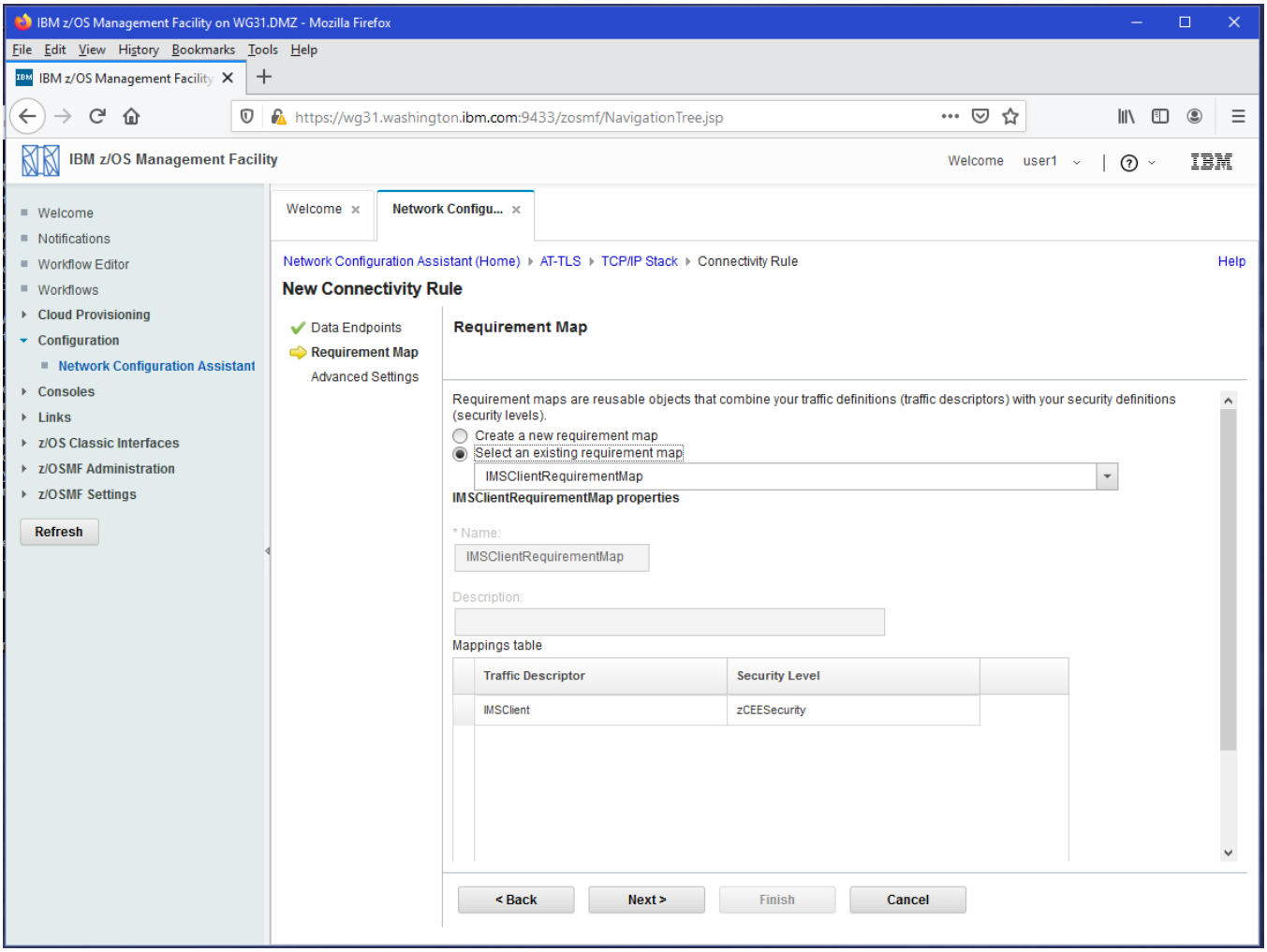
___36. Select the radio button beside *TCPIP1*. Use the *Actions* pull-down to select *Rules.* This is where these client definitions will be tied together. Use the *Actions* pull-down again and select *New* to create a new connectivity rule. Enter ***IMSClientRule*** for the *Connectivity rule name* and press **Next** to continue.






Mitch Johnson (mitchj@us.ibm.com)

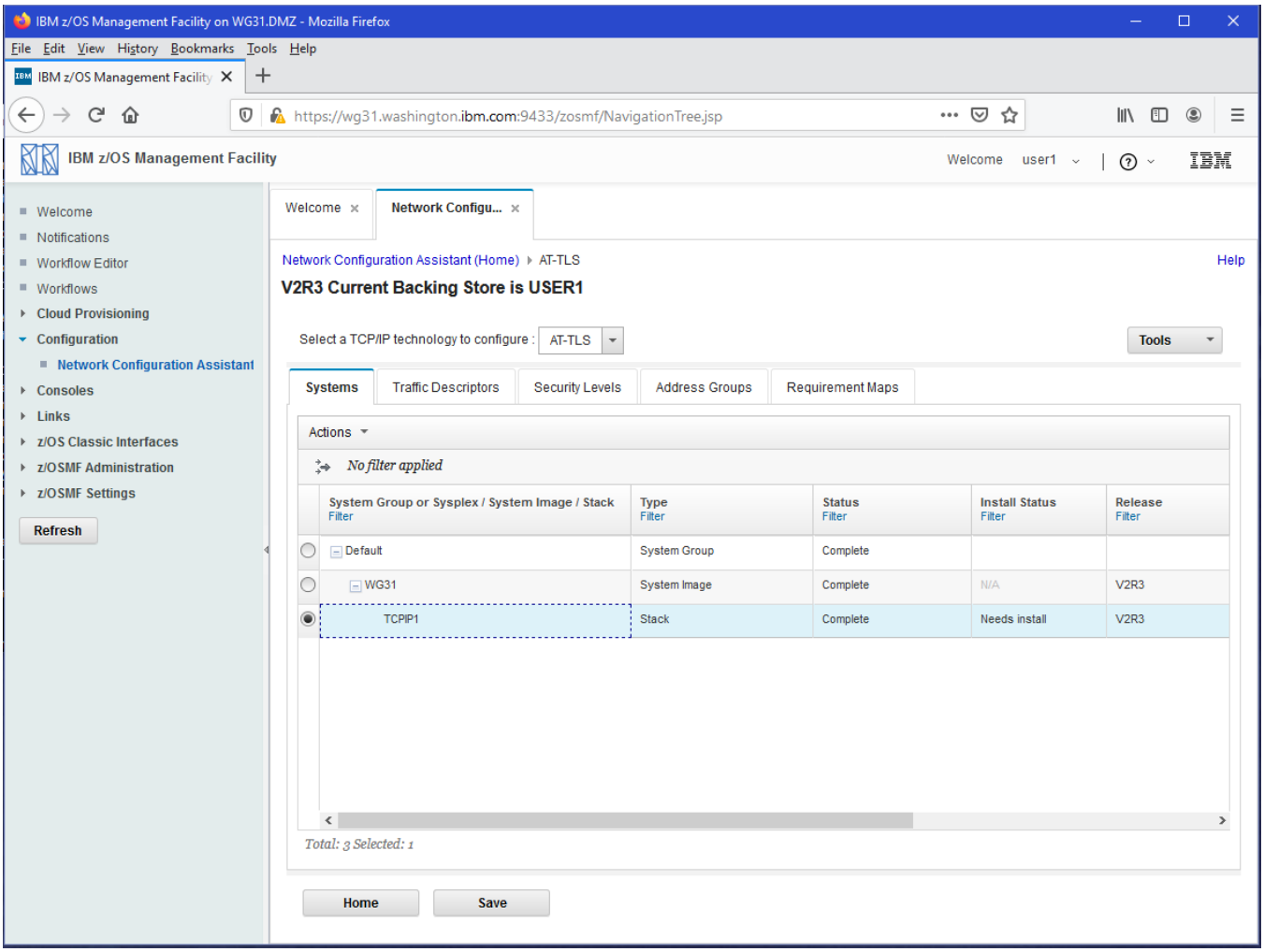
___37. On the *New Connectivity Rule – Requirement Map* window select the radio button beside *Select an existing requirement map*. Use the pull-down to select *IMSClientRequirementMap*. This should automatically populate the mapping table with *IMSClient* as the traffic descriptor and *zCEESecurity* as the security level. Press **Next** and then **Finish** to continue.






Mitch Johnson (mitchj@us.ibm.com)

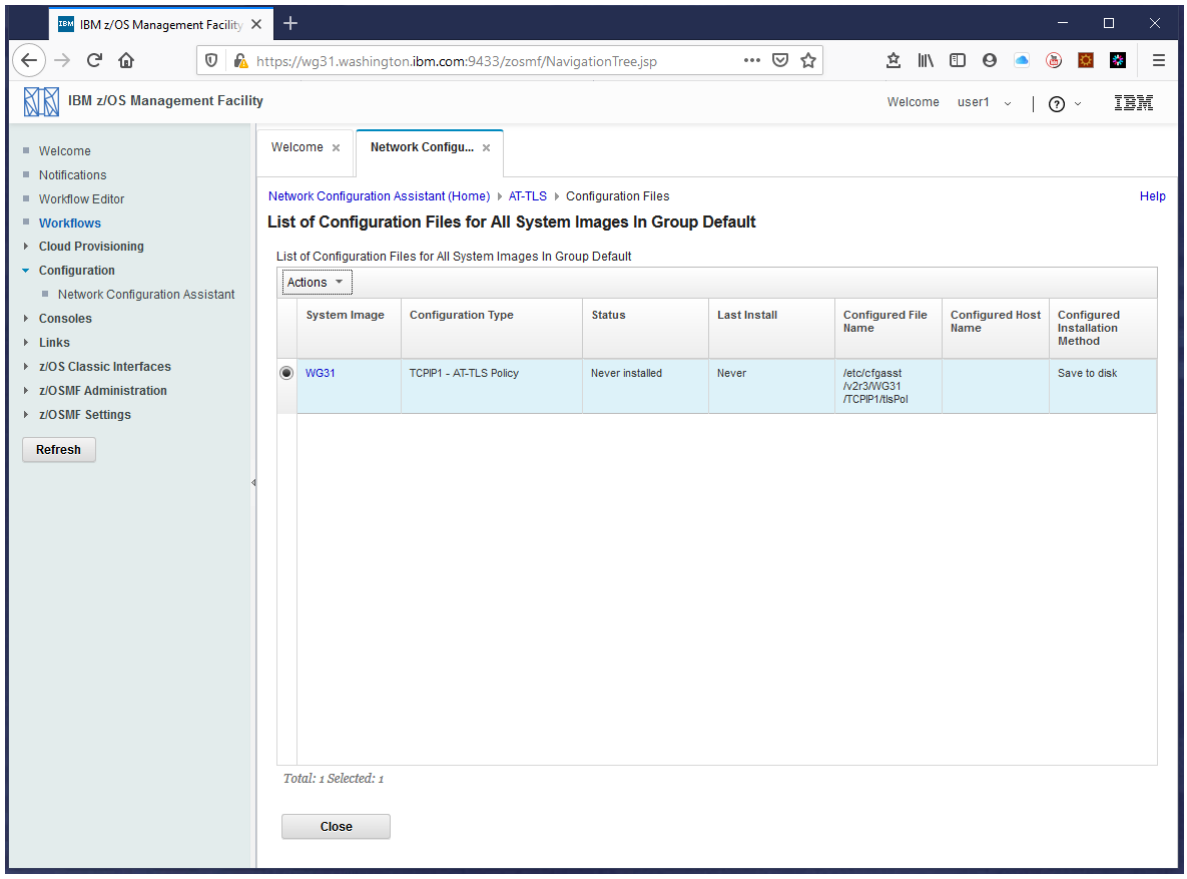
___38. Click **Close** to return the main screen. Select the radio button beside *TCPIP1* and use the *Actions* pull-down to select *Install All Files for This Group.*






Mitch Johnson (mitchj@us.ibm.com)

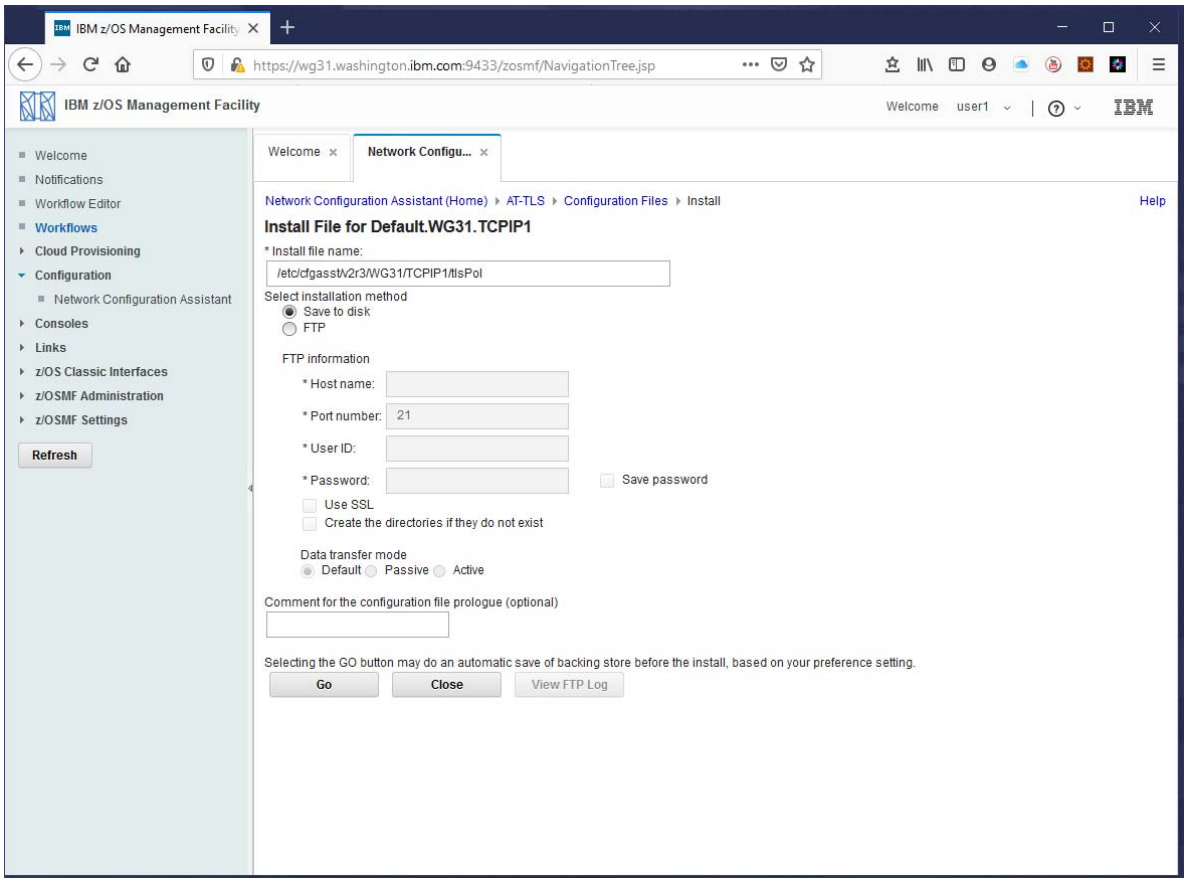
___39. On the *List of Configuration Files for All Systems Images in Group Default* window, select *WG31* and use the *Actions* pull-down to select *Install*.





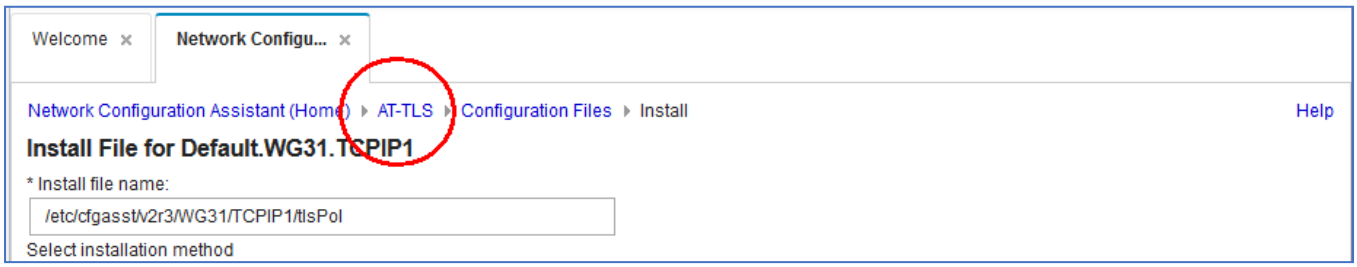

Mitch Johnson (mitchj@us.ibm.com)

___40. On the *Install File for Default.WG31.TCPIP1* window click the **GO** button to continue.




___41. Click **OK** twice to continue.

___42. Next click on *AT-TLS* as shown below to return to the primary window.




___43. 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 an active policy agent task (e.g. PAGENT).

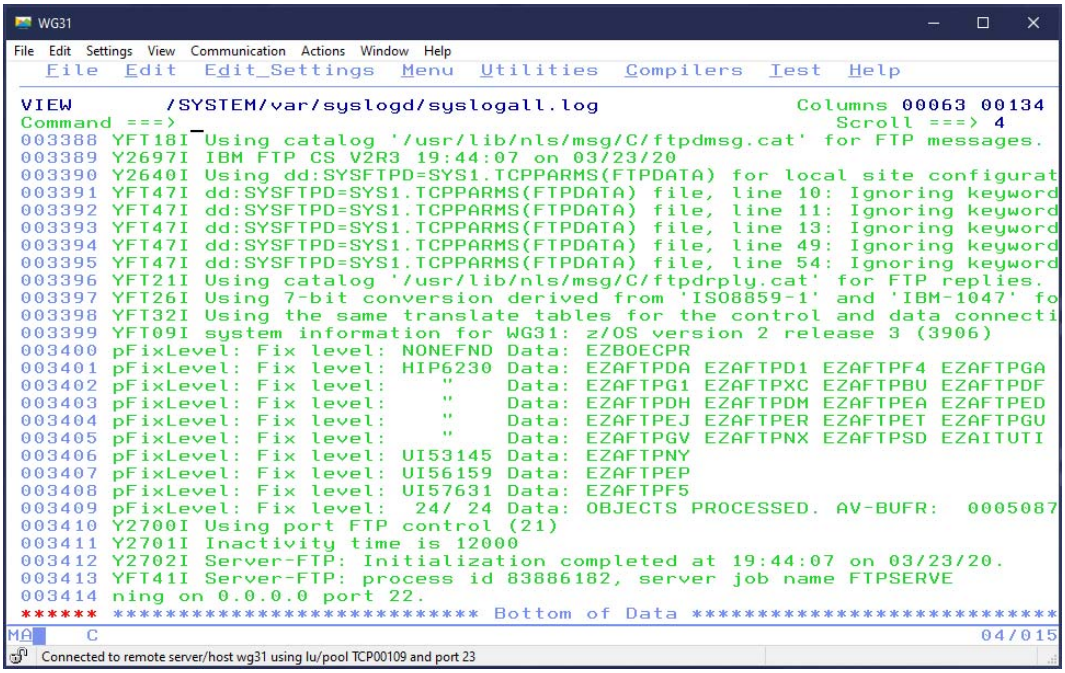
___1. This instance of the policy agent has been configured to use the *SYSLOGD* daemon task to log messages

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

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

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

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






Mitch Johnson (mitchj@us.ibm.com)

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

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

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

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

```
profile: Processing Image TTLS config file: '/etc/cfgasst/v2r3/WG31/TCPI
Processing:  'TTLSRule                          IMSConnectRule~1'
Processing:  'TTLSRule                          IMSClientRule~2'
Processing:  'TTLSGroupAction                   gAct1'
Processing:  'TTLSEnvironmentAction             eAct1~IMSConnect'
Processing:  'TTLSEnvironmentAction             eAct2~IMSClient'
Processing:  'TTLSConnectionAction              cAct1~IMSConnect'
Processing:  'TTLSConnectionAction              cAct2~IMSClient'
Processing:  'TTLSConnectionAdvancedParms       cAdv1~IMSConnect'
Processing:  'TTLSConnectionAdvancedParms       cAdv2~IMSClient'
Processing:  'TTLSKeyringParms                  keyR1'
Processing:  'TTLSKeyringParms                  keyR2'
Processing:  'IpAddrSet                         addr1'
Processing:  'PortRange                         portR1'
Processing:  'PortRange                         portR2'
profile: Finished processing Image TTLS config file
rocessing TTLS Group action 'gAct1'
rocessing TTLS Connection action 'cAct1~IMSConnect'
rocessing TTLS Connection action 'cAct2~IMSClient'
rocessing TTLS Environment action 'eAct1~IMSConnect'
rocessing TTLS Environment action 'eAct2~IMSClient'
cessing TTLS rule 'IMSClientRule~2'
cessing TTLS rule 'IMSConnectRule~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 configure 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) are

TCPCONFIG NOTTLS

___9. Stop and restart the server with MVS commands ***P BAQSTRT*** and ***S BAQSTRT***.

**Tech-Tip:** The server must be stopped and restart because there is an active session between the zCEE server and IMS Connect. The policies are not trigger until the socket session is restarted.



Mitch Johnson (mitchj@us.ibm.com)

### *Test the TLS connection from the zCEE Server to IMS*

Now use the REST client to test the API with AT-TLS enabled.

___1. Open a DOS command prompt and change to directory *c:/z/admin*.

___2. Enter the cURL commands below:

```
c:\z\admin>curl -X get --cacert certauth.pem --cert user1.p12:secret --cert-type P12
https://wq31.washington.ibm.com:9443/imdb/contact/LAST3
```

You should see the same results as before.

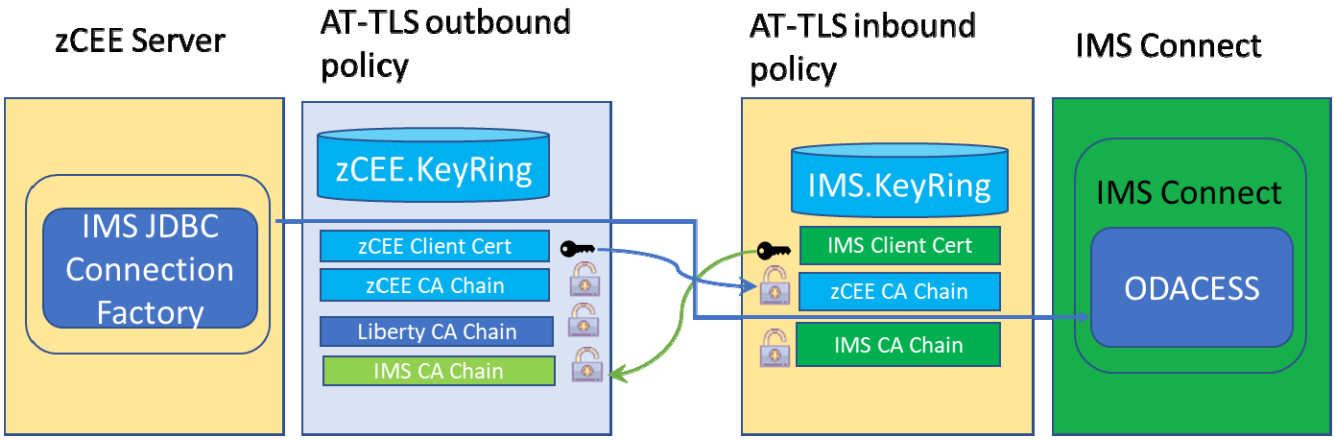
___3. View the *syslogd.log* file again and you should find the messages like the ones below. These messages are tracing the handshake between the outbound policy acting a client and the inbound policy acting as the server.

```
EZD1281I TTLS Map   CONNID: 00000A35 LOCAL: 192.168.17.220..2693 REMOTE :
192.168.17.220..5555 JOBNAME: BAQSTRT USERID: LIBSERV TYPE: OutBound STATUS:
Enabled RULE: IMSClientRule~2 ACTIONS: gAct1 eAct2~IMSClient cAc2~IMSClient

EZD1281I TTLS Map   CONNID: 00000A36 LOCAL: 192.168.17.220..5555 REMOTE :
192.168.17.220..2693 JOBNAME: IMS15HWS USERID: IMSSTC TYPE: InBound STATUS:
Enabled RULE: IMSConnectRule~1 ACTIONS: gAct1 eAct1~IMSConnect cAct1~IMSConnect

EZD1283I TTLS Event GRPID: 00000009 ENVID: 00000009 CONNID: 00000A36  R C:    0
Initial Handshake 000000501142EE90 0000005011429850 TLSV1.2  F0F0F3F5

EZD1283I TTLS Event GRPID: 00000009 ENVID: 00000008 CONNID: 00000A35  R C:    0
Initial Handshake 0000005011427EB0 0000005011422750 TLSV1.2  F0F 0F3F5
```

___4. Entering IMS Connect *VIEWHWS* command should show FRED and/or USER1 as the CLIENTIDs.

```
HWSC0001I    PORT=5555D    STATUS=ACTIVE     KEEPAV=0 NUMSOC=3
EDIT=        TIMEOUT=6000
HWSC0001I     CLIENTID USERID   TRANCODE DATASTORE STATUS      SECOND
CLNTPORT IP-ADDRESS          APSB-TOKEN
HWSC0001I     ODBB90D3 FRED           IVP15OOD  RECV          9
2735   192.168.017.220
HWSC0001I     ODBD40D8 USER1          IVP15OOD  RECV          424
2693   192.168.017.220
HWSC0001I     TOTAL CLIENTS=2  RECV=2 READ=0 CONN=0 XMIT=0 OTHER=0
```

The diagram below shows the flow of the request from the z/OS Connect server to IMS Connect. The server and IMS Connect are unaware of the involvement of the AT-TLS the TLS handshake and the fact that the message is encrypted between the two endpoints.




___5. There is information in the AT-TLS Knowledge Center for describing AT-TLS return codes (see URL https://www.ibm.com/support/knowledgecenter/SSLTBW_2.4.0/com.ibm.zos.v2r4.hald001/comtls.htm)
For example, if the return code had been 202 it would be caused by the explanation below.

> 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

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



Mitch Johnson (mitchj@us.ibm.com)

If the return code had been 414 and explanation can be found in the z/OS Cryptographic Services System SSL Programming (SC14-7495-40**)** manual or at URL
https://www.ibm.com/support/knowledgecenter/SSLTBW_2.4.0/com.ibm.zos.v2r4.gska100/sssl2msg1000885.htm

> **414 Certificate is not valid.**
>
> **Explanation:** Either the local certificate or the peer certificate is not valid.
>
> **User response:** Ensure that a valid certificate is being sent by the communication partner. Collect a System SSL trace containing a dump of the incorrect certificate and then contact your service representative if the error persists.

This is probably caused because the certificate authority certificate used to sign a server or personal certificate is not connect to the key ring. Or not properly connected to the key ring.

___6. The JCL in member *PASEARCH* in data set *USER1.ZCEE30.CNTL* can bee used to display AT-TLS policies. Submit this job and see a display of the contents of the policy agent's configuration file. Without the Configuration Assistant provided by z/OSMF this file would need to be maintained manually.

```
//BPXBATCH EXEC PGM=BPXBATCH,REGION=8M
//STDOUT  DD PATH='/tmp/paStd.out',
//           PATHOPTS=(OWRONLY,OCREAT),PATHMODE=SIRWXU
//STDERR  DD PATH='/tmp/paStd.err',
//           PATHOPTS=(OWRONLY,OCREAT),PATHMODE=SIRWXU
//STDPARM DD *
SH echo pasearch  -t | su
//COPY   EXEC PGM=IKJEFT01,DYNAMNBR=300
//SYSTSPRT DD SYSOUT=*
//PASTDOUT DD PATH='/tmp/paStd.out',PATHDISP=(DELETE,DELETE)
//PASTDERR DD PATH='/tmp/paStd.err',PATHDISP=(DELETE,DELETE)
//STDOUT   DD SYSOUT=*,DCB=(LRECL=1000,RECFM=V)
//STDERR   DD SYSOUT=*,DCB=(LRECL=1000,RECFM=V)
//SYSPRINT DD SYSOUT=*
//SYSTSIN  DD *
 OCOPY INDD(PASTDERR) OUTDD(STDERR)
 OCOPY INDD(PASTDOUT) OUTDD(STDOUT)
```

## *Optional*

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



Mitch Johnson (mitchj@us.ibm.com)

# Summary

In this step you have created AT-TLS inbound and outbound policies which protect the IMS Connect ODBA port of 5555. These policies respectively act as client and server surrogates for the z/OS Connect server and IMS Connect.



Mitch Johnson (mitchj@us.ibm.com)