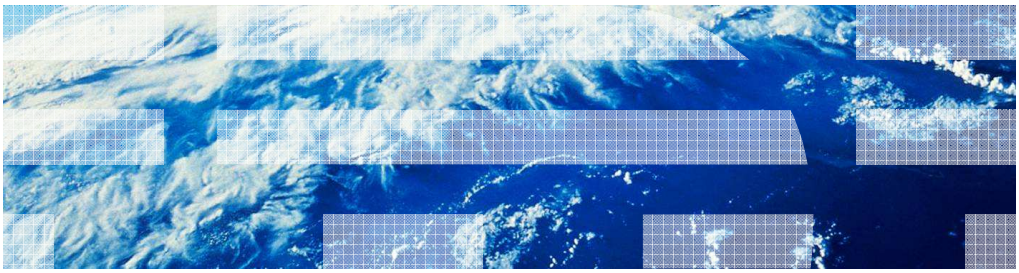
IBM WebSphere Application Server for z/OS V8

z/OS identity propagation exploitation



This presentation describes support for the z/OS Identity Propagation exploitation included in IBM WebSphere Application Server for z/OS V8.

Table of contents

This is the agenda of what is covered in this presentation. In the Overview, the problem is presented, and the way it had been solved in the past. It is highlighted how this feature is a better solution. Next, the main usage scenario for this feature is discussed. At the end, a summary is given of the advantages of this feature and how to use it. In the References, useful links are included to articles in the Information Center.

Section

Overview

The Overview section describes the problem being solved by this feature, how it was solved in the past, and the advantages of using this feature.

z/OS identity propagation exploitation - Overview (1 of 8)

The problem:

How are distributed identities handled on the mainframe?

This feature addresses the problem of how to handle distributed identities on the mainframe. Nowadays, customers have WebSphere Application Server on distributed platforms like AIX talking to WebSphere Application Server on the z/OS platform. The user repository is shared across all the Application Servers, but when the distributed identity arrives at z/OS customers want the z/OS security product to handle authorizing this identity, and when talking to a backend resource like CICS or DB2, the identity must be a z/OS SAF identity. WebSphere Application Server customers need a way of mapping the distributed identities to z/OS SAF identities.

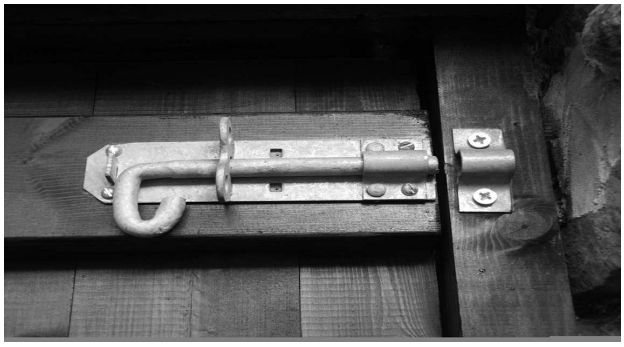
z/OS identity propagation exploitation - Overview (2 of 8)

- Background
- History:
  - Before V8.0, users can log in with a non-z/OS ID and a JAAS SAF mapping module in WebSphere Application Server mapped the distributed identity to a z/OS SAF ID
  - Any z/OS resources accessed outside of the Application Server container use the mapped z/OS SAF ID

In order to map distributed identities to z/OS SAF identities, WebSphere Application Server offered a limited solution before V8. A customer had to write a custom JAAS login module and add it to the Application Server security configuration in order to map the distributed identity to the z/OS SAF identity.

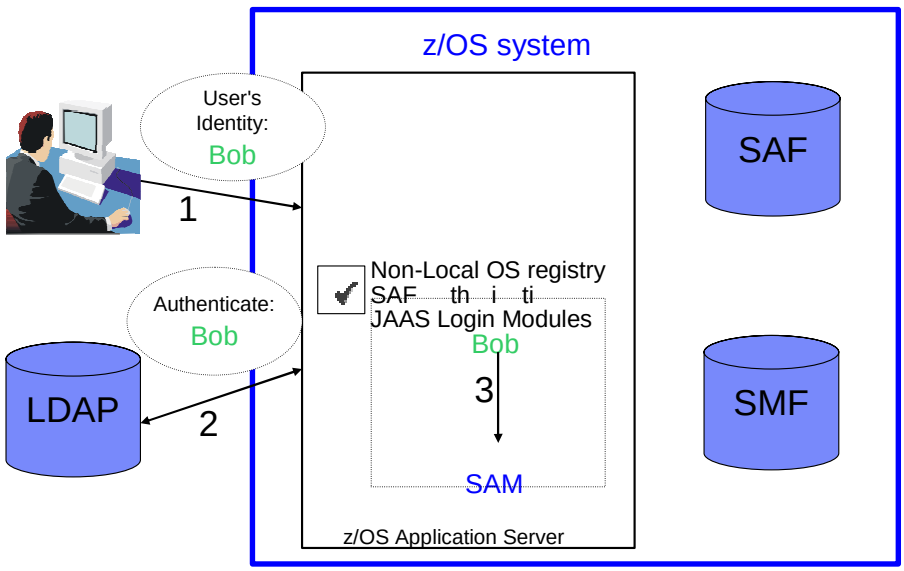
z/OS identity propagation exploitation - Overview (3 of 8)

- Limitations:
  - Auditability: The z/OS SMF Auditing system is not aware of the “original” distributed identity. Audit records only contain the mapped z/OS SAF ID.
  - Manageability: The mapping is handled by the WebSphere Administrator instead of the z/OS Security Administrator, which raises audit concerns for customers



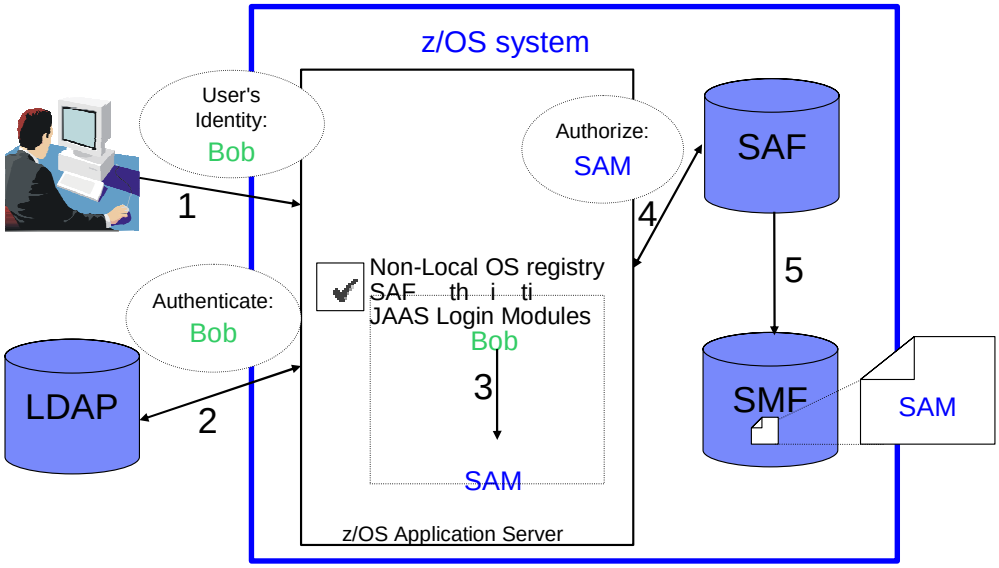
Using the JAAS login module solution has two limitations: auditability and manageability. Audit records generated by the z/OS SMF Auditing system will contain only the z/OS SAF identity- the original user, the distributed identity, is not tracked or audited. As far as manageability, the mapping is handled by the WebSphere administrator, instead of the z/OS security administrator.

z/OS identity propagation exploitation - Overview (4 of 8)



This diagram depicts a typical user scenario. WebSphere Application Server is deployed on the z/OS system using a non-Local OS user repository, in this case LDAP. The WebSphere Application Server security configuration is using SAF authorization, and the WebSphere Application Server administrator has configured a JAAS login module that will map the distributed identity of Bob to the z/OS SAF identity of SAM. In the first step, the user logs in with their LDAP identity of Bob. Secondly, Bob is authenticated against the LDAP registry. Thirdly, Bob is mapped to the z/OS SAF identity of SAM.

z/OS identity propagation exploitation - Overview (5 of 8)



In step 4, The SAF user SAM is authorized. At this point, the original identity, Bob, has been lost. When the z/OS SMF audit facility generates an audit record for authorizing SAM, the only information available is about the SAF identity.

z/OS identity propagation exploitation - Overview (6 of 8)

Problem:

How are distributed identities handled on the mainframe?

Solution:

Distributed identity propagation feature in SAF

The solution for handling distributed identities on the mainframe is to take advantage of the new feature in SAF: distributed identity propagation.

z/OS identity propagation exploitation - Overview (7 of 8)

- Distributed identity mapping and propagtion in z/OS 1.11:
  - new in V8
  - Provides a way for z/OS transactional subsystems to associate users' distributed identities with SAF user IDs under z/OS security control while maintaining the users' original identity information for audit purposes
  - Improves cross-platform interoperability and provides value for both host centric and heterogeneous application environments
- Benefits:
  - Auditability: The z/OS SMF Auditing system is aware of the “original” distributed user identity. Audit records contain both the distributed identity and the mapped z/OS SAF identity
  - Manageability: The identity mapping is handled by the z/OS Security Administrator, instead of the WebSphere Administrator

Being able to map and propagate distributed identities on z/OS is a SAF feature that was introduced in z/OS 1.11. This feature allows z/OS transactional subsystems, such as WebSphere and CICS, to associate a user's distributed identity with a SAF identity. The key advantages for this feature are auditability and manageability. When a distributed identity is mapped to a SAF identity using this feature, the z/OS SMF auditing subsystem does not lose track of the original user's identity, the distributed identity. Audit records will contain both identities. As far as manageability, the control for mapping distributed identities to SAF identities is now shifted to the z/OS security administrator, instead of the WebSphere administrator.

z/OS identity propagation exploitation - Overview (8 of 8)

This results in a more secure and accountable environment.

Section

Usage scenario

This section describes one usage scenario.

Usage scenario: Definition (1 of 3)

- WebSphere Application Server for z/OS Administrator wants to use SAF database for authorization with the non-Local OS user repository

The main usage scenario for exploiting the z/OS identity propagation feature involves being able use a non-Local OS user repository for authenticating users, but then use the SAF repository for authorization.

Usage scenario: Definition (2 of 3)

WebSphere Application Server for z/OS Administrator wants to use SAF database for authorization with the non-Local OS user repository

- Roles:
  - WebSphere administrator: configures security for WebSphere Application Server
  - z/OS security administrator: configures security for the z/OS systems and monitors SMF auditing
  - End user: logs in to the application hosted by the z/OS Application Server
- Goals:
  - When the distributed user logs into the application, the user is authenticated against the distributed registry, then mapped to a SAF identity and the authorization check is performed using the SAF identity.
  - Any audit records that are generated will contain both the distributed user and the mapped SAF identity

There are three roles in this usage scenario: first, the WebSphere administrator who is responsible for configuring security on the Application Server. The second role is the z/OS security administrator who is in charge of configuring security for the z/OS systems and monitors the SMF auditing records. Finally, the third role is that of the user who is logging into the application hosted by the z/OS application server. There are two goals in this usage scenario: first, the end user is able to log in using their distributed user id. The user is authenticated against the distributed user repository, and authorized against the SAF database, where it is mapped to a SAF identity. Secondly, the audit records that are generated for the end user logging in will contain both the distributed user and the mapped SAF identity.

Usage scenario: Definition (3 of 3)

WebSphere Application Server for z/OS Administrator wants to use SAF database for authorization with the non-Local OS user repository

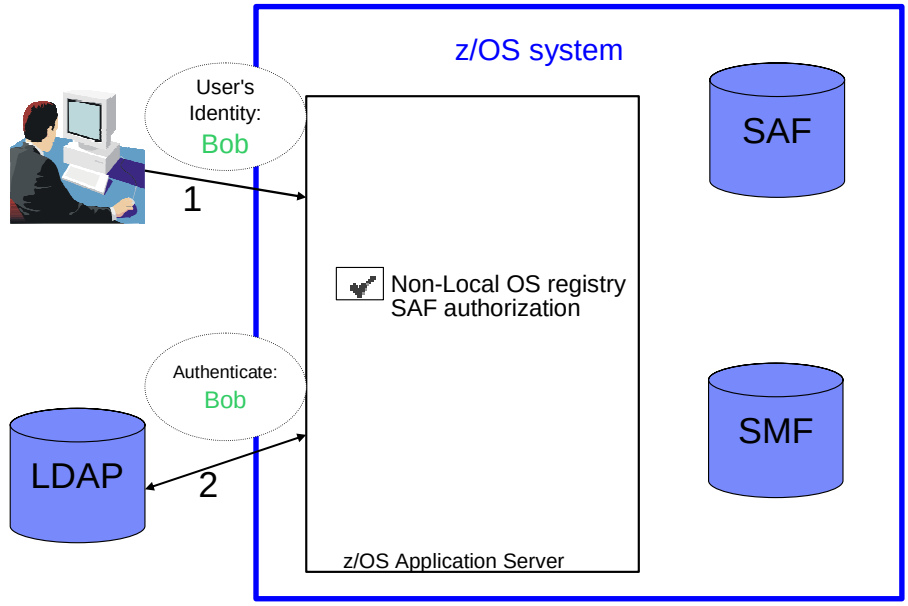
- Prerequisites:
  - z/OS 1.11 or later
  - WebSphere Application Server for z/OS V8 or later
- Configuration:
  - WebSphere administrator configures the application server to use a non-Local OS user registry and SAF authorization
  - z/OS security administrator configures filters in the SAF database to map the distributed users to SAF users. For example:

```
RACMAP ID(mstone1)  MAP
 USERDIDFILTER(NAME('cn=testuser,o=ibm,c=us'))
 REGISTRY(NAME('myLdapHost.ibm.com:389')) WITHLABEL('testuser to
 mstone1')
```


In order to complete this usage scenario, the z/OS system must be at 1.11 or later, and WebSphere Application Server for z/OS must be at V8 or later. There are certain configuration changes that need to be made. The WebSphere administrator configures the application server to use a non-Local OS user repository with SAF authorization. The z/OS security administrator configures filters in the SAF database for mapping the distributed users to SAF users. An example of the syntax for defining a filter is shown.

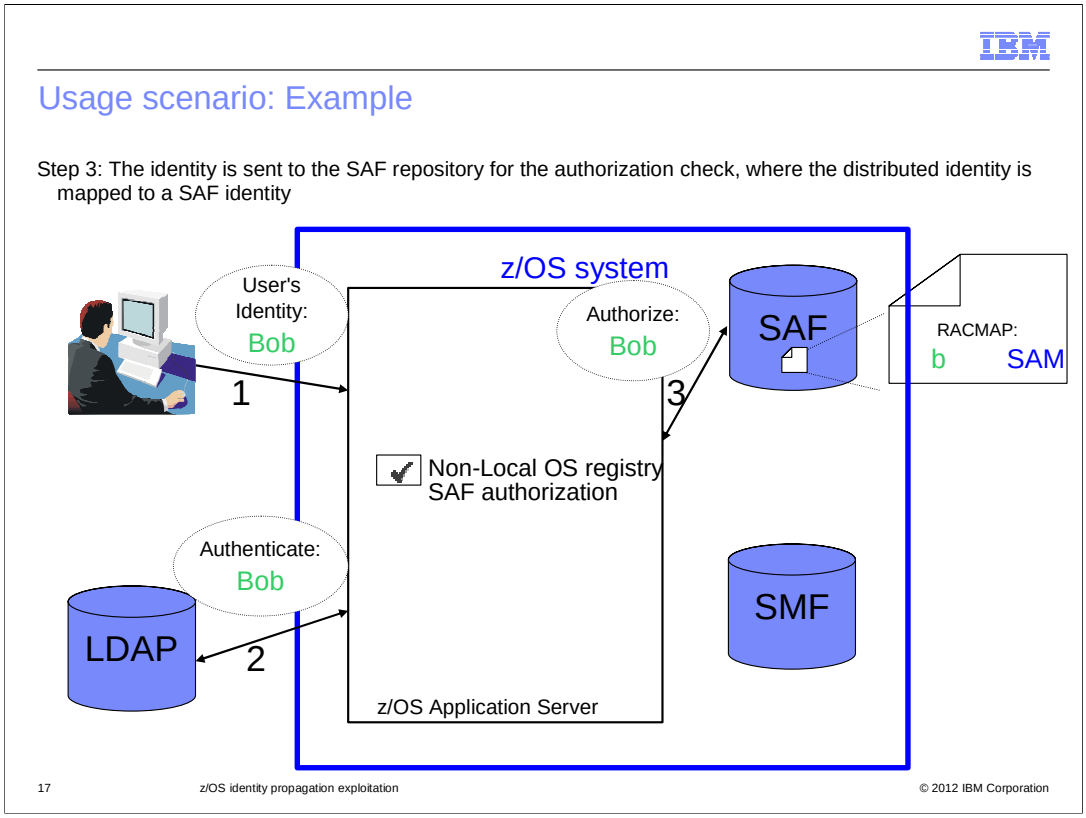
Usage scenario: Example

Steps 1 and 2: The distributed identity is authenticated against the LDAP user registry



16 z/OS identity propagation exploitation

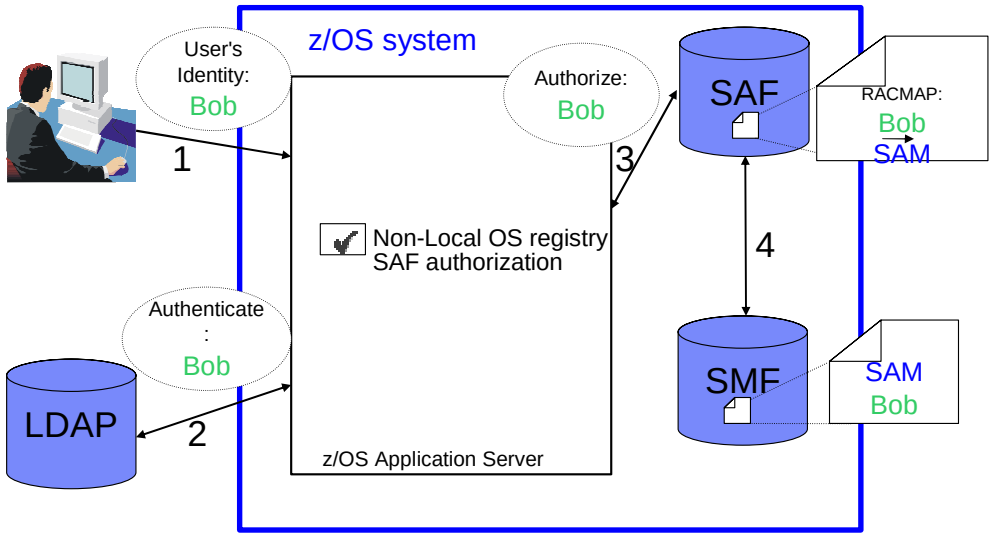
To illustrate this usage scenario, the same environment is used as the one described previously when presenting the problem. WebSphere is configured on a z/OS system with a distributed user repository for authentication, and SAF for authorization. In the first step, the user logs in to an application hosted on the Application Server by using their distributed identity, Bob. In step 2, Bob is authenticated in the LDAP repository. As one might notice, these first two steps remain identical to the problem scenario.



However, in step 3, the difference is now apparent. Before, a JAAS login module was configured to map the distributed identity to a SAF identity. Now, the SAF API to authorize the user can be called using the distributed identity of Bob. The filters defined by the z/OS security administrator in the SAF database then determine that Bob gets mapped to the z/OS SAF identity of SAM.

Usage scenario: Example

Step 4: An audit record is generated for the authorization check containing both the distributed identity and the z/OS identity



In the final step, an authorization check is made, and audit records the z/OS SMF Auditing subsystem contain both identities: Bob and SAM. Before, the record contained only SAM.

Section

Demonstration

This page signals the start of a live demonstration.

Skip to slide 30 after live demo.

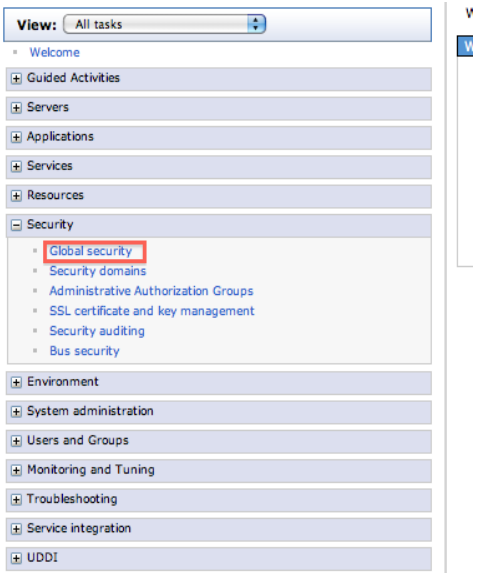
Usage scenario - Demonstration

WebSphere Application Server for z/OS administrator wants to use SAF database for authorization with the non-Local OS user repository

The main usage scenario for exploiting the z/OS identity propagation feature involves being able to use a non-Local OS user repository for authenticating users, but then use the SAF repository for authorization.

Demonstration: Configure the current realm to LDAP (1 of 4)

- **Step 1:** The *Global security* link is clicked on.



The user logs in to the administrative console, and the Global Security link is selected.

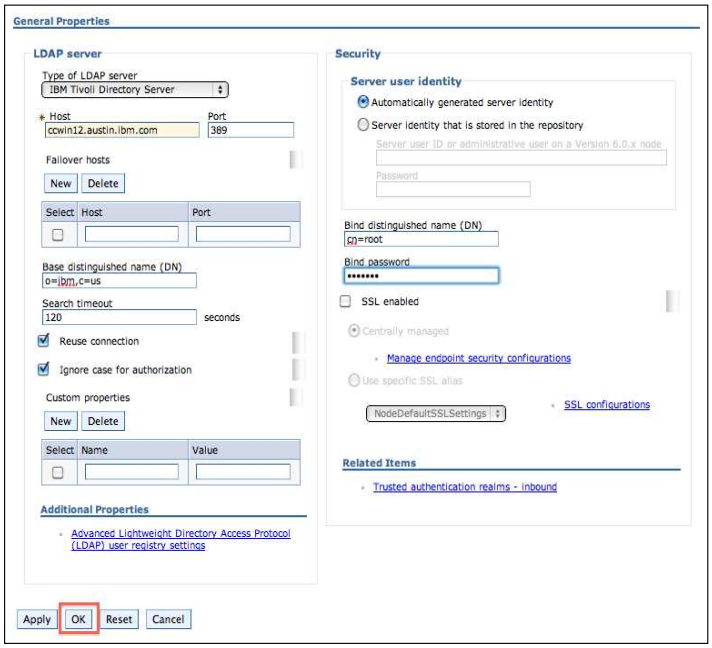
Demonstration: Configure the current realm to LDAP (2 of 4)

- **Step 2:** The stand-alone *LDAP registry* is selected from the dropdown under *Available realm definitions.* The *Configure* button is clicked

On the administrative console, the stand-alone LDAP registry is selected from the dropdown under “Available realm definitions”, then the Configure button is clicked.

Demonstration: Configure the current realm to LDAP (3 of 4)

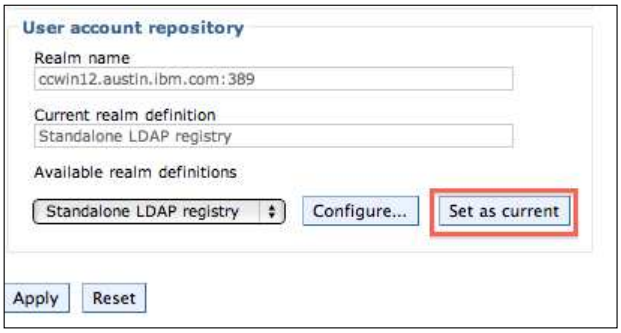
- **Step 3:** The required information is entered for the LDAP server, such as *Host* and *Port*, then *OK* is clicked*:*



On the LDAP server panel, the required information is entered for the LDAP server, and OK is clicked.

Demonstration: Configure the current realm to LDAP (4 of 4)

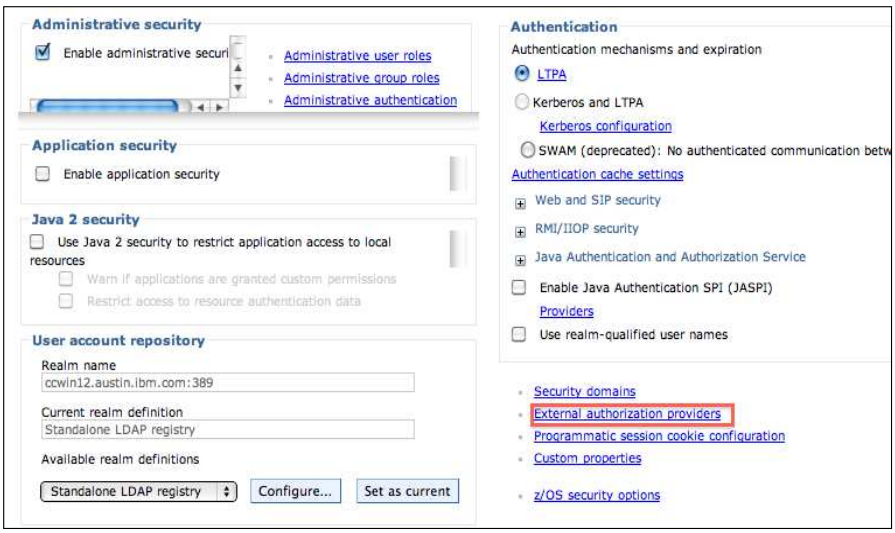
- **Step 4:** Back on the *Global Security* panel, the *Set as current* is clicked, then *Apply:*



On the Global Security panel, the button “Set as current” is clicked to switch to the LDAP user repository, then Apply is clicked.

Demonstration: Enable SAF authorization (1 of 3)

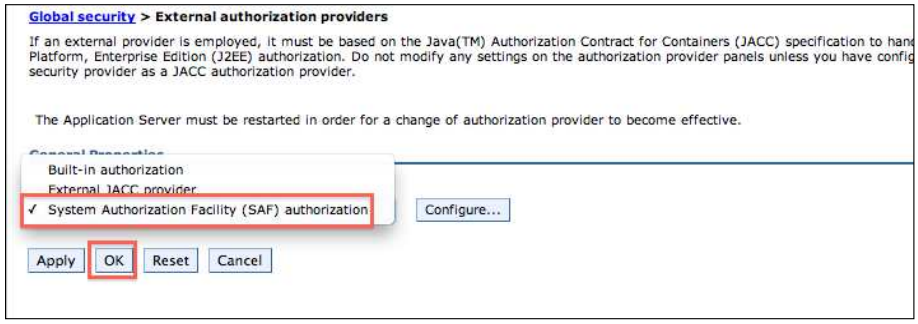
- **Step 1:** On the *Global security* panel, the *External authorization providers* link is clicked.



To use SAF authorization, on the Global Security panel, the “External authorization providers” link on the right is clicked.

Demonstration: Enable SAF authorization (2 of 3)

- **Step 2:** The *System Authorization Facility (SAF) authorization* is selected from the dropdown menu, and *OK* is clicked*:*



From the dropdown menu, “System Authorization Facility (SAF) authorization” is selected, then OK is clicked.

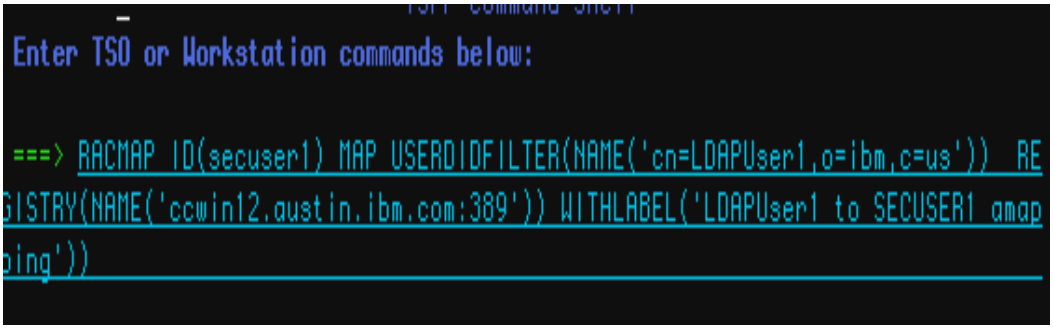
Demonstration: Enable SAF authorization (3 of 3)

- **Step 3:** Back on the *Global security* panel, *Apply* is clicked and the changes are saved*:*

Finally, on the Global Security panel, the Apply is clicked and the changes are saved.

Demonstration: Define filters in the SAF database to map distributed users to SAF users

- **Step 1:** For example, the LDAP user *LDAPUser1* is mapped to the SAF user *SECUSER1*:



On the z/OS system, the z/OS security administrator defines the mapping filters. The image here illustrates the syntax for mapping the LDAP user of LDAPUser1 to a SAF user of SECUSER1.

Demonstration: Configure the modified Snoop servlet

- Update the Snoop servlet to print out the SAF user ID.
  - Step 1: Expand the ear and war files to access the source code:
    cd $WAS_HOME/installableApps
    jar xvf DefaultApplication.ear
    jar xvf DefaultWebApplication_src.jar
- Step 2: edit SnoopServlet.java to add this code at the end:

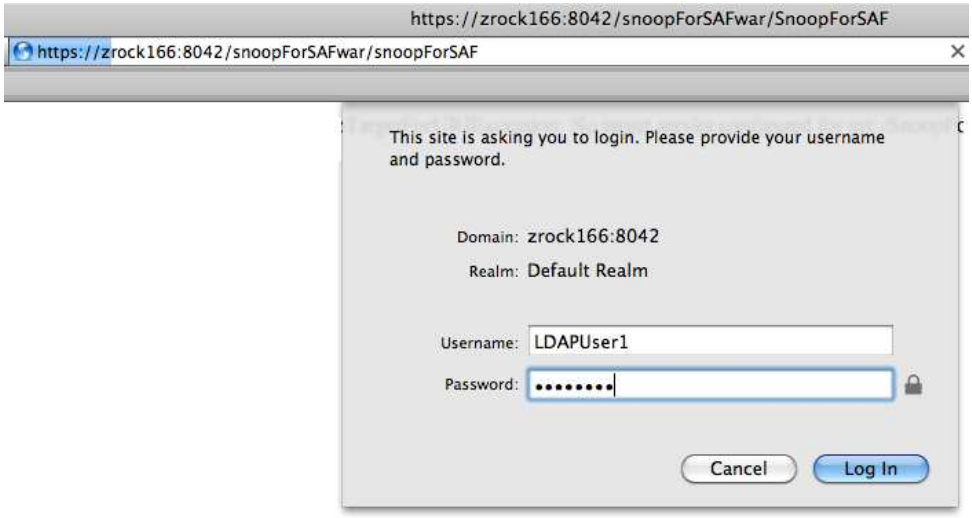
```
//get the caller subject
try{caller_subject = WSSubject.getCallerSubject();}
catch (WSSecurityException wse)
  {out.println("Caught excpetion getting the caller subject: "  + wse);}
if(caller_subject!=null)
  {String safUserIdCaller = WSSubject.getSAFUserFromSubject(caller_subject);
out.println("Caller SAF user id: "+ safUserIdCaller);}
```

- Step 3: Compile the source code and repackage the generated .class file into the war, then package the war into the ear file
- Step 4: Install the modified DefaultApplication.ear application.

For testing, the Snoop servlet is updated to print out the SAF user ID of the distributed user who logged in to the servlet. The servlet's source code is shipped with the WebSphere Application Server product. In Step1, the application ear file is expanded using the jar command, and then the war file within that ear is expanded. In Step 2, the file titled SnoopServlet.java is modified to add the code above as indicated. In Step 3, the .java file is compiled and a new class file is generated. This class file is repackaged into the .war file, and the .war file is repackaged into the .ear file. In Step 4, the modified application DefaultApplication.ear is installed on the application server.

Demonstration: Run the test application (1 of 2)

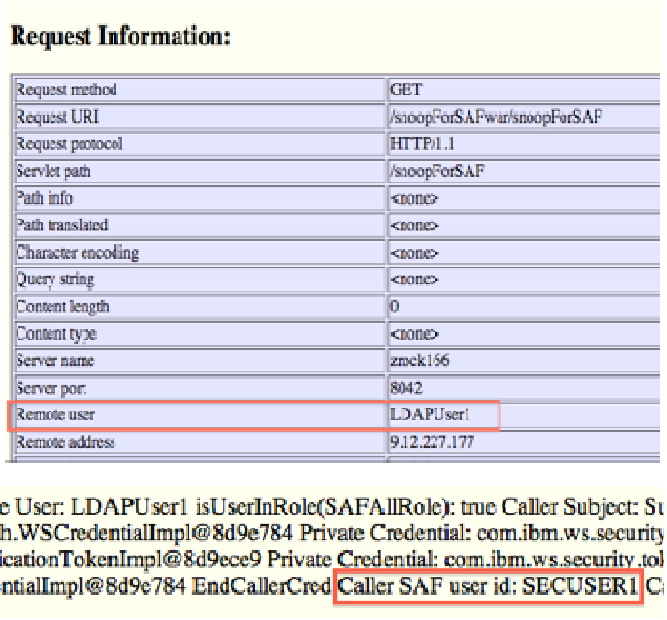
- **Step 1:** WebSphere application server is restarted.
- **Step 2:** A web browser URL is set to the updated snoop app, and the user logs in with *LDAPUser1*



In Step 1, the Application Server is restarted. In Step 2, the URL for the updated snoop application is accessed. When prompted for a user and password, the distributed user LDAPUser1 and its password are entered.

Demonstration: Run the test application (2 of 2)

▪**Step 3:** The expected output is verified. The remote user is the distributed user ID, LDAPUser1. The mapped SAF user ID will be SECUSER1.



Request Information:

Request method	GET
Request URI	/snoopForSAFwar/snoopForSAF
Request protocol	HTTP/1.1
Servlet path	/snoopForSAF
Path info	<none>
Path translated	<none>
Character encoding	<none>
Query string	<none>
Content length	0
Content type	<none>
Server name	znock166
Server port	8042
Remote user	LDAPUser1
Remote address	9.12.227.177

User Principal: LDAPUser1 Remote User: LDAPUser1 isUserInRole(SAFAllRole): true Caller Subject: Su
Credential: com.ibm.ws.security.auth.WSCredentialImpl@8d9e784 Private Credential: com.ibm.ws.security
com.ibm.ws.security.token.AuthenticationTokenImpl@8d9ece9 Private Credential: com.ibm.ws.security.tol
com.ibm.ws.security.auth.WSCredentialImpl@8d9e784 EndCallerCred Caller SAF user id: SECUSER1 C

31 z/OS identity propagation exploitation

In the output, the remote user is LDAPuser1, and the mapped SAF user is SECUSER1.

Section

Summary

This section provides a summary of what was discussed in this presentation.

Summary

- Using the distributed identity mapping feature replaces the need to configure JAAS mapping modules, places more control in the hands of the z/OS security administrator, and provides a complete end-to-end auditing solution for z/OS SMF.
- The flagship scenario for this feature involves configuring WebSphere with a non-Local OS user registry and enabling SAF authorization
- The demonstration goes over this scenario in detail
- Other scenarios exist in the Information Center, for mapping distributed identities in other forms (such as asserted DN, certificates, Kerberos users) to SAF identities

The advantage of using the z/OS identity propagation feature is that the WebSphere administrator does not need to configure any JAAS mapping modules. Instead, the control is given to the z/OS security administrator for defining mapping filters in the SAF database. And more importantly, the audit records in the z/OS SMF product will contain both the distributed user ID and the SAF user id. The most common usage scenario to take advantage of this feature involves a non-Local OS user repository configured with SAF authorization. The demonstration goes over how to configure this scenario in detail. Furthermore, there are other usage scenarios for using the distributed identity propagation feature that are documented in more detail in the Information Center.

References

- Information Center: Using distributed identity mapping for SAF

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=matt&product=was-nd-mp&topic=tsec_use_identity_saf

- Information Center: Distributed identity mapping using SAF

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=matt&product=was-nd-mp&topic=csec_identity_saf

- Information Center: Distributed identity filters configuration in z/OS security

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=matt&product=was-nd-mp&topic=rsec_config_identity_filters

This slide contains links to useful information.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WASV8z_IdentityPropagation.ppt

This module is also available in PDF format at: ../WASV8z_IdentityPropagation.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, disclaimer, and copyright information