IBM TRIRIGA Application Platform
3.7.0

Single Sign-on User Guide

Note

Before using this information and the product it supports, read the information in “Notices” on page 49.

This edition applies to version 3, release 7, modification 0 of IBM® TRIRIGA® Application Platform and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Authenticating users by using SSO

To gain access to IBM TRIRIGA applications, a user must be authenticated as a valid user of the system and must be granted permission to access applications and functions in the IBM TRIRIGA suite of applications. Many customers use single sign-on (SSO) authentication to manage access by their users to multiple applications in their environment.

Chapter 2. Types of authentication

The TRIRIGA Application Platform uses its own native authentication by default.

With native authentication, the user enters their user name and password in an IBM TRIRIGA login screen. The TRIRIGA Application Platform authenticates the user by comparing the user name and password that the user entered with the user name and password that are stored in the IBM TRIRIGA database.

Single sign-on authentication, which is not native to IBM TRIRIGA, can also be used for authentication. With single sign-on authentication, the user logs in to applications with a user name and password that are stored in an existing Lightweight Directory Access Protocol (LDAP) directory or Active Directory. The application server or web server authenticates the user by comparing the user name and password that the user entered with the user name and password that are stored in the directory server.

Chapter 3. Requirements for and limitations of SSO requests in TRIRIGA Application Platform

In an SSO environment, the user name and password that the user enters must match the user name and password that are stored in the directory server. The application server or web server then authenticates the user and inserts the user name into the HTTP request header.

The user name in the HTTP request header must exactly match the user name that is stored in the IBM TRIRIGA database. When configured properly, IBM TRIRIGA reads the user name from the request header and internally authenticates it against the IBM TRIRIGA database.

IBM TRIRIGA supports the following methods of inserting the user name into an HTTP header:

- Remote User - The web server or application server authenticates the user and puts the user name in the REMOTE_USER HTTP header. The Java™ call is `request.getRemoteUser()`.
- User Principal - The web server or application server authenticates the user and puts the user name in the special UserPrincipal HTTP header. The Java call is `request.getUserPrincipal().getName()`.
- HTTP Header - The web server or application server authenticates the user and puts the user name in a specific named HTTP header attribute.

In addition to the insertion methods, IBM TRIRIGA supports several options for the user name after it is retrieved from the HTTP header:

- Removal of Domain Name - In some SSO environments, the LDAP Domain Name is provided along with the user name, however, only the *username* portion is configured in the IBM TRIRIGA database. If the full string in the HTTP header is provided in the form of *MyCompany\username*, enabling this feature strips *MyCompany* or the domain portion from *username*.
- Case Sensitivity - Some directory servers supply the user name in a mixed case, depending on a number of conditions. By default, IBM TRIRIGA user names are case-sensitive. If it is determined that the directory server is providing user names with mixed cases, you can disable the case-sensitive check in the SSO process.

Considerations:

- If you are using a web server to provide the authentication portion, disable the HTTP port on the application server after the web server configuration completes. Keeping the application server's HTTP port open might create a vulnerability point. If the HTTP port is not disabled and the user goes to that port, the user is prompted for their credentials and the user name and password are verified in the IBM TRIRIGA database.
- IBM TRIRIGA is compatible with SSO when SSO is configured properly. After the appropriate IBM TRIRIGA properties are enabled for SSO, IBM TRIRIGA can accept tokens that are provided by properly configured application servers with SSO. IBM Support can assist with configuring IBM TRIRIGA properties for SSO. However, due to the number of supported products, technologies, and configurations that are supported by IBM TRIRIGA, IBM Support cannot help with the configuration of SSO within your environment.
- If you enable SSO and attempt to launch a UX application with the non-SSO URL, you will not be automatically logged in through SSO nor get a login screen. You will be presented with a message indicating that you "Cannot sign into IBM TRIRIGA as you do not have a valid user." Make sure that you are using the SSO URL to access the UX application.

Limitations:

- IBM TRIRIGA does not support **Security Assertion Markup Language** (SAML) or credential-less login mechanisms such as SmartCard or Common Access Card (CAC) as a method of authentication for its non-browser clients.
  - Unsupported non-browser clients include the following clients:

- IBM TRIRIGA CAD Integrator/Publisher
- IBM TRIRIGA Reservation (VSTO) add-in for Microsoft Outlook

– SSO solutions must provide a mechanism for **Basic Authentication** for non-browser clients. SAML and SmartCard or CAC do not support Basic Authentication for non-browser based clients.
– The best practice if you are using SAML or SmartCard/CAC is to authenticate directly to IBM TRIRIGA on a separate process server or integration server as opposed to the SSO enabled application server. This solution requires users to use their IBM TRIRIGA user name and password to sign in.
– An alternative best practice is to set up a separate non-SAML SSO solution for non-browser client users, which can support **Basic** or **NTLM Authentication**. This solution requires SmartCard/CAC users to use their SmartCard/CAC user name and password to sign in.

- Effective April 2020, Microsoft will continue to disable **Basic Authentication** for newly created tenants by default. Starting October 2020, Microsoft will begin to disable **Basic Authentication** in tenants that have no recorded usage. However, Microsoft decided to postpone the disabling of **Basic Authentication** with Microsoft 365/Exchange Online services for those tenants still actively using it until the second half of 2021. As an alternative, Microsoft will support the **OAuth** open standard to connect to Exchange Online services.
- Due to authentication requirements, the upload process no longer batches uploads when you use **NTLM Authentication** with IBM TRIRIGA CAD Integrator/Publisher. Also, the upload process for NTLM no longer provides progressive messages, for example, `Uploaded 10 of 100 spaces`. The upload process message for NTLM now displays `Uploading spaces`. This change does not impact the progressive feedback of other authentication schemes.
- IBM TRIRIGA Advanced Room Search does not support Identity Provider (IdP) initiated SSO. It only supports Service Provider (SP) initiated SAML, OAuth, or OpenID Connect (OIDC) SSO.

Chapter 4. How SSO works

Many possible configurations can insert the user name into the HTTP header. Configurations on a reverse proxy web server, configurations at the application server layer, or various authentication plug-ins at each of those layers can insert the user name into the HTTP header.

In general, the process occurs in the following order.

- The user enters the web server URL in a browser or accesses the application by using a client.
- The user might be prompted to enter a user name or password or seamless sign-on might occur. Seamless sign-on, where the server does not challenge the browser or client, is not supported in some configurations.
- The web server, application server, or authentication plug-in verifies the information with the authentication source.
- If the login is successful, the web server appends the user credentials to the HTTP header and sends them to the application server.
- The application server processes the user credentials and logs in the user to the application.

Note: In the IBM TRIRIGA Workplace Reservation Manager application, if you click a link such as the Building link in the **Find Room/Resource** dialog, a browser instance opens in a new window and you are prompted to sign in. The sign in request occurs because of security constraints; the session and login configuration cannot be shared between Outlook and the browser.

Chapter 5. Configuring IBM TRIRIGA with SSO

If you have a web server that is set up with single sign-on authentication, you can determine whether those credentials can be used to sign on to IBM TRIRIGA.

Procedure

1. Configure your web server for reverse proxy access to the application server. For configuration details, see the documentation that is provided by your application server provider.
2. After the web server and application server are communicating by using reverse proxy, enter the following URL in your web browser: `http://web_server/context_path/html/en/default/admin/requestTest.jsp`.

   The web page shows the HTTP headers that are passed from the web server to the application server.
3. On the application server, set the properties in the `TRIRIGAWEB.properties` file based on the SSO variables that are returned at the URL. By default, the `TRIRIGAWEB.properties` file is in the `Tririga/config` folder.

If...	Then, set the following properties.
The results show Remote User set with the login.	**SSO=**Y **SSO_REMOTE_USER=**Y Set all other SSO properties to N.
The results show UserPrincipal set with the login.	**SSO=**Y **SSO_USER_PRINCIPAL=**Y Set all other SSO properties to N.
If the results show user name on a header, make note of the header name, for example, *OTHER_SSO_USER_NAME*.	**SSO=**Y **SSO_REQUEST_ATTRIBUTE_NAME=** *OTHER_SSO_USER_NAME* Set all other SSO properties to N.

4. Restart the application server so that the changes take effect.

Chapter 6. SSO properties in IBM TRIRIGA

Several properties control an IBM TRIRIGA SSO configuration.

The SSO properties are in the `TRIRIGAWEB.properties` file. By default, the `TRIRIGAWEB.properties` file is in the `Tririga/config` folder of the application server. The application server must be restarted before the property value changes take effect.

Property	Options	Default	Description
SSO	N, Y	N	If set to Y, the environment runs in single sign-on (SSO) mode.
SSO_BACKING_SERVER_PORT	*number*	`-1`	The port number that is used by the back-end server. If the SSO server port does not match the back-end server port, this property must be set. If `-1` or any other negative value is set for this property, then the port number that is set for the front-end server is also set for the back-end server port.
SSO_DISABLE_UNAUTHORIZED _STATUS	N, Y	N	The `unauthorized.jsp` page sends an `HTTP Error 401` response in the HTTP Header. If set to Y, the header response is disabled. If you want the `HTTP Error 401` response sent, set this property to N.
SSO_REMOTE_USER	N, Y	Y	If set to Y, the `request.getRemoteUser()` method is used to sign in. The user name must exactly match the user name that is created in IBM TRIRIGA. When the value of **SSO_USER_PRINCIPAL** is Y, set **SSO_REMOTE_USER** to N.
SSO_REMOVE_DOMAIN_NAME	N, Y	Y	If set to Y, the prefixed or appended domain name is removed from the directory server user name that is passed by using the **SSO_REMOTE_USER** property. • If user names contain a domain name when passed from the directory server and user names in IBM TRIRIGA contain only the user name, set this property to Y. • If user names contain a domain name when passed from the directory server and user names in IBM TRIRIGA include the domain name, set this property to N.

Property	Options	Default	Description
SSO_REQUEST_ATTRIBUTE_NAME	[headername], sm_user, [username], [$WSRU]	*headername*	The name of the property that is inserted into the HTTP header whose value is the IBM TRIRIGA user name. The value can be blank. **Example 1:** For use with SiteMinder, set **SSO_REQUEST_ATTRIBUTE_NAME**=sm_user **Example 2:** For use with WebSphere® Application Server standalone or WebSphere Application Server Liberty, set **SSO_REQUEST_ATTRIBUTE_NAME**=$WSRU to pull the header from the plugin. On the IHS or Apache server, be sure to use **HEADER UNSET $WSRU** to be sure the header is only set at the web server or application server layer. **Tip:** This property will take priority over **SSO_REMOTE_USER** and **SSO_USER_PRINCIPAL**. Make sure the value of **SSO_REQUEST_ATTRIBUTE_NAME** is blank if you use **SSO_REMOTE_USER**=Y or **SSO_USER_PRINCIPAL**=Y. This property is case sensitive. Use the requestTest.jsp page to check the correct parameter name. When not in use, it must be set to a non-blank value. If the user name is stored in a distinct HTTP attribute variable, set **SSO_REMOTE_USER** to N, and set this property to the HTTP attribute name. In some systems, you can define the variable name in which the user name is located. In this case, set this property to the variable name in your system.
SSO_USER_PRINCIPAL	N, Y	N	If the system is configured to append the User Principal Name (UPN) to the HTTP header, set this property to Y. If set to Y, the HTTP header parameter **UserPrincipal** is used, and the user name is retrieved by calling the `request.getUserPrincipal().getName()` method. When the value is Y, set the value of the **SSO_REMOTE_USER** property to N.
USERNAME_CASE_SENSITIVE	N, Y	Y	If set to Y, sign-in user names are case-sensitive. If you want to authenticate without case sensitivity, set this property to N.

Some Java Applets prompt for the Windows user name and password, which is a known security issue with the Java plug-in and SSO. Affected applets might include: Brava! Document Viewer, Gantt, Association Viewer, and Workflow Expression Editor. Enter the SSO user name and password again to gain access to these applets.

Chapter 7. Forcing users to log in through SSO

If you want to force users to log in through SSO, you must prevent them from using the default login page. Provide an alternative login page that does not contain a user name, password, or login button.

Procedure

1. Add the following properties to the TRIRIGAWEB.properties file to specify the alternative login page and directory.

Property	Values	Description
ALTERNATE_INDEX_HTML	*File name*	The file name of the alternative sign-in page, for example, index.html.
ALTERNATE_RESOURCE_DIRECTORY	N, Y	The path to the alternative sign-in page resource directory, for example, C:*pathToTRIRIGA*\userfiles\alt.

2. Restart the application server.

Chapter 8. Troubleshooting SSO

Several issues are known to occur with single sign-on, for example, if it is not configured properly.

Invalid user name or password error.

Make sure the SSO settings in the `TRIRIGAWEB.properties` file are set and the application server is restarted.

The user name is case-sensitive in IBM TRIRIGA. To see the actual user name that is passed from the web server to IBM TRIRIGA, open the following address in a browser: `http://web_server/html/en/default/admin/requestTest.jsp`.

You can find the user name in the **Request Parameters** section, in the **Header Parameters** section next to **`getUserPrincipal`**, or in both sections.

Map labels are shown only in English.

If Esri map labels are shown in English even though your user profile is using a different language, the **`SSO_BACKING_SERVER_PORT`** property in the `TRIRIGAWEB.properties` file might not be configured for the internal non-SSO port.

HTTP requests are no longer forwarded to IBM TRIRIGA.

After you upgrade IBM TRIRIGA on WebSphere Application Server, IBM HTTP Server no longer forwards requests to IBM TRIRIGA.

You must reconfigure the web server in WebSphere Application Server. One method of reconfiguring the web server is to use the **WebSphere Customization Toolbox** (WCT). WCT contains the **Web Server Plug-ins Configuration Tool**, which steps through the process of deleting and re-creating the web server definition for IBM HTTP Server.

Tip: When you specify the application server location in the **Configuration Scenario Selection** dialog, if your configuration scenario is local then browse to the location of the `\AppServer` folder. For example, a common location for the application server is `C:\Program Files (x86)\IBM\WebSphere\AppServer`.

If you change the host name, check the `plugin-cfg.xml` file to make sure that it contains the correct host name specified. Specifically, check the **`Transport Hostname`** property. The `plugin-cfg.xml` is typically in *`pathToInstall`*`/IBM/HTTPServer/Plugins/config/webServerName/plugin-cfg.xml`.

Note: For more information, see Troubleshooting SSO in our IBM TRIRIGA wiki.

Chapter 9. Examples of configuring SSO

The steps that you take to set up SSO for the TRIRIGA Application Platform depend on several factors, such as the application server on which TRIRIGA Application Platform is installed, as well as the web server and directory server in the environment.

About this task

This section provides requirements and configurations for some of the most commonly-used SSO combinations. **Single sign-on** refers to the ability to have a single set of credentials that use a directory server for multiple applications. By definition, single sign-on is not the same as **seamless sign-on**, which might not challenge a user for credentials during the access process.

Since we cannot possibly know of every SSO combination in use, use these configurations as a guide for configuring your SSO environment with TRIRIGA. For support information, see the **Web Server & Third-Party Server Compatibility** section of the **IBM TRIRIGA Compatibility Matrix**.

Note: The SSO content in this section was written for TRIRIGA Application Platform 3.4.2 and above. While some of these configurations might work with releases prior to 3.4.2, it is best to follow the SSO documentation for the TRIRIGA Application Platform release that you are using.

Configuring SSO for TRIRIGA on WebSphere Liberty with IIS and AD

There are several steps for configuring single sign-on (SSO) with WebSphere Application Server Liberty, Microsoft Internet Information Services (IIS), and Microsoft Active Directory (AD).

Contents

I. Installing WebSphere Application Server web server plug-in on IIS

Install the WebSphere Application Server web server plug-ins and the WebSphere Customization Toolbox: **WAS Supplements 8.5.5**. The plug-ins are available in the WAS Supplements package for WebSphere Application Server on Passport Advantage. As with the WebSphere Application Server installation, you use the IBM Installation Manager to install the web server plug-ins and the WebSphere Customization Toolbox.

For details on obtaining the WAS Supplements 8.5.5 package from Passport Advantage, see: Part numbers of WebSphere software used by IBM TRIRIGA (http://www-01.ibm.com/support/docview.wss?uid=swg21692375). Download WAS Supplements 8.5.5 for your Platform. Then install the fix packs for the supplements.

Note: You do **not** need to install the WebSphere Application Server standalone. However, you do need to install the WebSphere Customization Toolbox as the next step requires it to generate the web server folder. The web server folder will contain the web server plug-in as well as the configuration files. IIS will point to the web server folder as a resource, but WAS standalone does not need to be running.

For details on how to install and generate the plug-in on the IIS server, see: Adding a plug-in configuration to a web server.

II. Configuring WebSphere Liberty property *trustedSensitiveHeaderOrigin*

There was a change on WebSphere Liberty 19.0.0.4 that added a new configuration property named *trustedSensitiveHeaderOrigin*.

See reference: Potential WebSphere Application Server problems when deployed behind a WebSphere-aware proxy server

On WebSphere Liberty, *trustedSensitiveHeaderOrigin* is configured as a *HttpDispatcher* custom property. This property has a default value of "none", which means that a subset of WebSphere-specific HTTP headers will not be trusted from any host. The property also accepts value a of "*" (all), or a comma-separated list of IP addresses. For a secure deployment in which proxy servers are used, the *trustedSensitiveHeaderOrigin* property should be configured with a comma-separated list of IP addresses corresponding to those of any WebSphere-aware proxy servers in front of the WebSphere server.

Alternatively, to enable the original unsecured behavior, set *trustedSensitiveHeaderOrigin="*"*, which will direct the WebSphere server to trust all headers sent from any host or proxy. This value must only be used for testing, or if the WebSphere server is isolated from external connections.

For WebSphere Liberty servers, add the following line to the server.xml:

```
<httpDispatcher trustedSensitiveHeaderOrigin="<TRUSTED_PROXY_IP_ADDRESS>"/>
```

Replace <TRUSTED_PROXY_IP_ADDRESS> with the IP of the web server machine where the WebSphere plug-in is installed.

See reference: HTTP Dispatcher (httpDispatcher)

III. Configuring IIS to pass web requests to WebSphere Liberty

About this task

If you are using an already installed web server plug-in on the web server, reconfigure it to use the web server plug-in by using the following procedure.

Note: If the web server plug-in is not installed, then install it, but do not use the following procedure since it is completed automatically during web server plug-in installation. You need to complete the steps below only if you are reconfiguring IIS Version 8.x to use an existing web server plug-in.

You can create a web server configuration beforehand by using the WebSphere Customization Toolbox. The configuration will be installed to `C:\Program Files (x86)\IBM\WebSphere\Plugins\config\webserver1`. When you are prompted for the web server details, point to the URL by using the remote installation.

Procedure

1. On the Server Manager screen, click Tools > Internet Information Services (IIS) Manager. This action starts the IIS application, and creates a new virtual directory for the website instance that you intend to use with WebSphere Liberty. These instructions assume that you are using the default website.
2. Expand the tree until you see Default Web Site.
3. Right-click Default Web Site and select Add Virtual Directory to create the directory with a default installation.
4. On the Virtual Directory Alias window, enter `sePlugins` in the Alias field.
5. In the Physical Path field of the Web Site Content Directory window, browse to and select the `pluginConfigFolder\bin` directory and click OK. For example, select `C:\Program Files (x86)\IBM\WebSphere\Plugins\config\webserver1\bin`.
6. Click Test Settings. If the settings test fails, then either change the permissions of the physical directory, or select Connect As, and let IIS connect as a Windows user account that has authority to files in that physical path.

Attention: When you click Test Settings, you might encounter the following warning message if you use the default Pass-thru authentication setting: "`Cannot verify access to path`". For more information, see the Microsoft documentation on this subject.

7. Click OK to add the sePlugins virtual directory to your default website.
8. In the navigation tree, select the sePlugins virtual directory that you created.
9. On the Features panel, double-click Handler Mappings, and then click Edit Feature Permissions on the Actions panel.
10. Select a resource.
11. Select Script and Execute, if they are not already selected then click OK.
12. Manually copy the plug-in binaries file, `iisWASPlugin_http.dll`, to the `plugins_root\bin\` directory. For example, copy the plug-in binary files to the `C:\Program Files\IBM\WebSphere\Plugins\bin\` directory.
13. Return to the IIS Manager window, and expand the Web Sites folder in the left-hand navigation tree of that window.
14. Select Default Web Site in the navigation tree.
15. Add the Internet Services Application Programming Interface (ISAPI) filter into the IIS configuration.
16. On the Default Web Site Properties panel, complete the following steps.
    a) Double-click the ISAPI Filters tab.
    b) Click to open the Add/Edit Filter Properties dialog.
    c) Enter `iisWASPlugin` in the Filter name field.
    d) Click Browse to select the plug-in file, `iisWASPlugin_http.dll`, located in the `root\config\webserverName\bin` directory.
    e) Click OK to close the Add/Edit Filter Properties dialog.
    f) An sePlugins filter is automatically created here and on the server node. They can both be removed.
17. In the navigation tree, select the top level server node.
18. On the Features panel, double-click ISAPI and CGI Restrictions, and then, on the Actions panel, click Add.
19. To determine the value to specify for the ISAPI or CGI Path property, browse to and select the same plug-in file that you selected in the previous step. For example, `root\config\webserverName\bin\iisWASPlugin_http.dll`.
20. Enter `WASPlugin` in the Description field, select Allow extension path to execute, and click OK to close the ISAPI and CGI Restrictions dialog.
21. Set the value in the `plugin-cfg.loc` file to the location of the configuration file at `plugins_root\config\webserver_name\plugin-cfg.xml`.
    - The default location of `plugin-cfg.loc` is `C:\Program Files\IBM\WebSphere\Plugins\bin\IIS_webserverName`. You can also create the file manually in a text editor. The only item that the `plugin-cfg.loc` file contains is the path to the `plugin-cfg.xml` file. For example, `C:\Program Files\IBM\WebSphere\Plugins\config\webserverName\plugin-cfg.xml`.
    - The default location of `plugin-cfg.xml` is `C:\Program Files\IBM\WebSphere\Plugins\config\IIS_webserver1\plugin-cfg.xml`. The location varies depending on how you have configured your system. If the web server and WebSphere Liberty are on separate machines, you have a remote installation.
    - If the web server and WebSphere Liberty are on the same machine, then you have a local installation, and the correct location of the configuration file might be set. For example, `C:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml`.
    - If the two servers are on the same machine, and the application server is federated, you have a local distributed installation. For example, `C:\IBM\WebSphere\AppServer\profiles`

`\custom01\config\cells\dmgrcell\nodes\managed_node\servers \webserver1\plugin-cfg.xml`.

22. Generate a new version of `plugin-cfg.xml` from your Liberty application server. As the user that is running Liberty (may be Administrator) go to the Java home directory that Liberty uses, and run: jconsole. Connect to your server then click the MBeans tab. Under the WebSphere section, locate the `com.ibm.ws.jmx.mbeans.generatePluginConfig` MBean `generateDefaultPluginConfig` operation to generate the `plugin-cfg.xml` file, or call the`generatePluginConfig` operation to customize installation root directory and server name before you generate the `plugin-cfg.xml` file.
23. You will need to copy the `plugin-cfg.xml` file to the web server, replacing the existing `plugin-cfg.xml` file configured for IIS to read. Here is an example of the generated `plugin-cfg.xml`:

```
<?xml version="1.0" encoding="UTF-8"?><!--HTTP server plugin config file for defaultServer
generated on 2015.03.02
at 20:29:36 GMT-->
<Config ASDisableNagle="false" AcceptAllContent="false"
AppServerPortPreference="HostHeader" ChunkedResponse="false"
FIPSEnable="false" IISDisableNagle="false" IISPluginPriority="High"
IgnoreDNSFailures="false" RefreshInterval="60"
ResponseChunkSize="64" SSLConsolidate="false" TrustedProxyEnable="false"
VHostMatchingCompat="false">
   <Log LogLevel="Error" Name=".\logs\webserver1\http_plugin.log"/>
   <Property Name="ESIEnable" Value="true"/>
   <Property Name="ESIMaxCacheSize" Value="1024"/>
   <Property Name="ESIInvalidationMonitor" Value="false"/>
   <Property Name="ESIEnableToPassCookies" Value="false"/>
   <Property Name="PluginInstallRoot" Value="."/>
<!-- Configuration generated using httpEndpointRef=defaultHttpEndpoint-->
<!-- The default_host contained only aliases for endpoint defaultHttpEndpoint.
     The generated VirtualHostGroup will contain only configured web server ports:
        webserverPort=80
        webserverSecurePort=443 -->
   <VirtualHostGroup Name="default_host">
      <VirtualHost Name="*:81"/>
   </VirtualHostGroup>
   <ServerCluster CloneSeparatorChange="false" GetDWLMTable="false"
IgnoreAffinityRequests="true" LoadBalance="Round Robin"
Name="defaultServer_default_node_Cluster" PostBufferSize="0" PostSizeLimit="-1"
RemoveSpecialHeaders="true" RetryInterval="60">
      <Server CloneID="861158e3-af1e-47fe-b1c4-f21622bbc450" ConnectTimeout="5"
ExtendedHandshake="false" MaxConnections="-1"
Name="default_node_defaultServer" ServerIOTimeout="900" WaitForContinue="false">
         <Transport Hostname="triqaapp11.tivlab.usnv.ibm.com" Port="8001" Protocol="http"/>
         <Transport Hostname="triqaapp11.tivlab.usnv.ibm.com" Port="8443" Protocol="https">
            <Property Name="keyring" Value="keyring.kdb"/>
            <Property Name="stashfile" Value="keyring.sth"/>
            <Property Name="certLabel" Value="LibertyCert"/>
         </Transport>
      </Server>
      <PrimaryServers>
         <Server Name="default_node_defaultServer"/>
      </PrimaryServers>
   </ServerCluster>
   <UriGroup Name="default_host_defaultServer_default_node_Cluster_URIs">
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/*"/>
   </UriGroup>
   <Route ServerCluster="defaultServer_default_node_Cluster"
UriGroup="default_host_defaultServer_default_node_Cluster_URIs"
VirtualHostGroup="default_host"/>
</Config>
```

IV. Configuring SSO with Microsoft IIS

About this task

After configuring IIS to pass web requests to WebSphere Liberty, the next step is to set up SSO.

Procedure

1. On the application server, set the following attributes in the `TRIRIGAWEB.properties` file. This file should be located in the `Tririga/config` folder.

```
SSO=Y
SSO_REMOTE_USER=N
SSO_REMOVE_DOMAIN_NAME=Y
SSO_REQUEST_ATTRIBUTE_NAME=$WSRU
USERNAME_CASE_SENSITIVE=N
```

If AD contains usernames with inconsistent cases (for example, if TRIRIGA users have been entered in lower case, and users in AD are in mixed cases), use the following setting to turn off the case-sensitive check upon login:

```
USERNAME_CASE_SENSITIVE=N
```

If you want to force users to log in through SSO, you must prevent them from using the default login page. Provide an alternative login page that does not contain a username, password, or login button. Use the following settings to specify the alternative login page and directory:

```
ALTERNATE_INDEX_HTML=<indexFileName.html>
ALTERNATE_RESOURCE_DIRECTORY=/<pathToTRIRIGA>/userfiles/alt
```

2. Restart the application server to use the new settings.
3. On the IIS server, right-click My Computer and select Manage.
4. Expand Services and Applications. Select and expand Internet Information Services (IIS) Manager.
5. Under IIS, expand Web Sites.
6. Right-click Default Web Site and select Properties.
7. In the Default Web Site Properties panel, select the Directory Security tab. In Authentication and access control, click Edit.
8. In the Authentication Methods panel, uncheck the box next to Enable anonymous access.
9. Select the type of authentication. If you do not know which one to set, do not choose all three. Try checking one at a time, restarting IIS after the change, and testing to see if the setting works correctly. The correct setting depends on your company's security setup.
   - Integrated Windows authentication
   - Digest authentication for Windows domain servers
   - Basic authentication (password is sent in clear text)
10. Enter the domain name in Default Domain and Realm. The fields that are available depend on the check boxes selected in Authenticated access.
11. Click OK.
12. Restart IIS and make sure that the application server has been restarted.
13. Make sure you have a login within TRIRIGA that matches your domain login. If your domain login is john.doe, the user name in the profile of the TRIRIGA Employee record should be john.doe.

    Note: Logins are case-sensitive. Some logins in Active Directory are stored in mixed case.
14. Open your browser to `http://<<webserver>>/`. It should take you directly to TRIRIGA.

V. Troubleshooting IIS and Maximum File Upload Size

About this task

If you have `MAXIMUM_UPLOAD_FILE_SIZE_MEGABYTES` property set to a large value (for example, 50 MB) in the `TRIRIGAWEB.properties` file, but you are still running into problems with uploading large files, Microsoft IIS Web Server's default configuration needs to be changed to allow large files to be uploaded too.

Procedure

1. Open IIS Admin, click on the server, and then double-click Configuration Editor.

2. After that opens, select `system.webServer/security/requestFiltering` from the dropdown at the top of the window.
3. Expand "requestLimits" and edit the "maxAllowedContentLength" setting to the desired amount.

   Note: The default is 30000000 bytes, which is roughly 28 MB. In this example, the value was changed to 90000000 bytes, which is roughly 84 MB. IIS has a maximum size of 2 GB, which is 2147483648 bytes.

Configuring SSO for TRIRIGA on traditional WebSphere with IHS and TDS

There are several steps for configuring single sign-on (SSO) with traditional WebSphere Application Server, IBM HTTP Server (IHS), and Tivoli Directory Server (TDS).

Contents

I. Setting up IBM HTTP Server and WebSphere Application Server web server plug-in

Two things are required to have IBM HTTP Server working with WebSphere Application Server: (1) the IBM HTTP Server itself and (2) the WebSphere Application Server web server plug-in. The WebSphere Application Server package includes the IBM HTTP Server and the web server plug-in. Or you can obtain the plug-in from the Supplements package for WebSphere Application Server on Passport Advantage. As with the WebSphere Application Server installation, you use the IBM Installation Manager to install the web server plug-in.

For details on obtaining the Supplements package from Passport Advantage, see: Part numbers of WebSphere software used by IBM TRIRIGA (http://www-01.ibm.com/support/docview.wss?uid=swg21692375).

Use the following notes for guidance with the IBM HTTP Server installation:

- Install as a root user.
- IBM HTTP Server installation instructions can be found in the installation folder (readme > InstallGuide).

  Note: The installer requires X Window System.
- The web server plug-in installation is part of the IBM HTTP Server installation. Do not forget to install the plug-in.
- The installation path is usually: `/opt/IBM/HTTPServer`.
- After you finish installing IBM HTTP Server, configure the administrative server in: `/opt/IBM/HTTPServer/conf/admin.conf`.

```
# Port used to access the Administration Server
Listen @LISTENINGPORT@
# Default user and group settings for the server
User @USER@
Group @GROUPNAME@
# ServerName gives the name and port that the server uses to identify
# itself. This can often be determined automatically, but
# specifying it explicitly can prevent problems during startup.
#
# If this is not set to a valid DNS name for your host, server-generated
```

```
# redirections do not work. See also the UseCanonicalName directive.
#
# If your host does not have a registered DNS name, enter its IP
# address. You must access it by its address anyway, and this makes
# redirections work correctly.
#
ServerName hostname:@@AdminPort@@
```

- Set the IBM HTTP Server administrator password.

  Note: If the administrator password does not exist, you must first run **htpasswd** with the **-c** option to create the password.

  1. Switch to the HTTP`Server_installdir`/bin directory on your machine.
  2. To set the administrator password, enter the following command based on your operating system:
     - In AIX, Linux, or Solaris, enter: **./htpasswd -b ../conf/admin.passwd user password**.
     - In Windows, enter: **htpasswd -b conf\admin.passwd user password**.
- After the IBM HTTP Server installation is completed, start the web server administration server by using the following commands:

```
<ibm_HTTP_server_path>/bin/adminctl start
<ibm_HTTP_server_path>/bin/apachectl start
```

II. Configuring traditional WebSphere property *trustedSensitiveHeaderOrigin*

There was a change on traditional WAS 9.0.0.11 that added a new configuration property named *trustedSensitiveHeaderOrigin*.

See reference: Potential WebSphere Application Server problems when deployed behind a WebSphere-aware proxy server

On traditional WebSphere, the property is configured as an HTTP channel custom property. This property has a default value of "none", which means that a subset of WebSphere-specific HTTP headers will not be trusted from any host. The property also accepts value a of "*" (all), or a comma-separated list of IP addresses. For a secure deployment in which proxy servers are used, the *trustedSensitiveHeaderOrigin* property should be configured with a comma-separated list of IP addresses corresponding to those of any WebSphere-aware proxy servers in front of the WebSphere server.

Alternatively, to enable the original unsecured behavior, set *trustedSensitiveHeaderOrigin="*"*, which will direct the WebSphere server to trust all headers sent from any host or proxy. This value must only be used for testing, or if the WebSphere server is isolated from external connections.

For traditional WAS, set trustedSensitiveHeaderOrigin as a custom property of HTTP channel.

See reference: HTTP transport channel custom properties

III. Configuring IHS to pass web requests to WebSphere Application Server

About this task

After you install IBM HTTP Server, you configure it to forward requests to the application server.

The following steps demonstrate how to configure the web server by using the WebSphere Customization Toolbox.

Procedure

1. Start the WebSphere Customization Toolbox.
2. In the Web Server Plug-in Configuration box, select Create.
3. In the Web Server Selection window, select IBM HTTP Server.

4. In the Web Server Configuration File Selection dialog, select the existing IBM HTTP Server `httpd.conf` file and port.
5. Optionally, in the Setup IBM HTTP Server Administration Server dialog, configure an administrative server to administer the web server and create an administrative user ID and password.
6. Optionally, set up the IBM HTTP Server Administration Server as a Windows Service.
7. In the Web Server Definition Name dialog, specify a unique web server definition, such as the default `webserver1`.
8. In the Configuration Scenario Selection dialog, specify the location of the application server. If your configuration scenario is local, browse to the location of the `\AppServer` folder. For example, a common location for the application server is `C:\Program Files (x86)\IBM\WebSphere \AppServer`.
9. In the WebSphere Application Server Profile Selection dialog, select the WebSphere Application Server profile to configure with the current web server plug-in. For example, `AppSrv01`.
10. In the Plug-in Configuration Summary dialog, review the items you chose and select Configure.

IV. Configuring SSO with IBM HTTP Server

About this task

After you configure IBM HTTP Server to forward web requests to WebSphere Application Server, you set up SSO with IBM HTTP Server.

Procedure

1. On the application server, set the following attributes in the `TRIRIGAWEB.properties` file. This file should be located in the `Tririga/config` folder.

```
SSO=Y
SSO_REMOTE_USER=Y
SSO_REMOVE_DOMAIN_NAME=Y
SSO_REQUEST_ATTRIBUTE_NAME=sm_user
```

2. Configure IBM HTTP Server to use LDAP by editing the `httpd.conf`.
   a) Activate the LDAP module by adding the following line:
      - `LoadModule ibm_ldap_module modules/mod_ibm_ldap.dll`
      - On Unix operating systems, the module name will be different:
      - `LoadModule ibm_ldap_module modules/mod_ibm_ldap.so`
   b) Define the Location. Here is an example:

```
<Location "/">
   LdapConfigFile "/opt/IBM/HTTPServer/conf/ldap.prop"
   AuthName "Directory server"
   AuthType Basic
   Require valid-user
</Location>
```

   c) Definitions:
      - `Location`: Defines a scope for the action. In this case, we are authenticating against the scope of the web application.
      - `LdapConfigFile "/opt/IBM/HTTPServer/conf/ldap.prop"`: Defines the path to your LDAP configuration file.
      - `AuthName "Directory server"`: Defines the text that appears in the popup which prompts the user for their login and password.
      - `AuthType Basic`: Defines that Basic is being used as the authentication method. The login and password are encoded and included in the headers of each request sent by the browser.

- `Require valid-user`: Defines which users can access the resources. In this case, any authenticated user has access. It is possible to define a list of user names or groups. See the next step.

3. Configure the LDAP settings by editing the `ldap.prop` file.

   Below is an example of a minimal configuration file for a TDS server over SSL.

   An `ldap.prop.sample` file is available in the `conf` directory of IHS that you can rename to `ldap.prop` and customize for you environment. This file contains other options as well as explanations regarding the possible values of these configuration options.

   Make sure each line reflects your configuration. For example, for `ldap.URL`, substitute `tdsserver.company.com` with the name of your directory server, as well as the port (`sslport`), organizational unit (ou), organization (o), etc. Make sure to update the path to the keyfile and stash file.

```
ldap.realm=ldaprealm
ldap.URL=ldap://tdsserver.company.com:sslport/ou=organization,o=company.com?mail
ldap.group.URL=ldap://tdsserver:sslport/ou=groups,o=company.com
ldap.transport=SSL
ldap.application.authType=none
ldap.user.authType=BasicIfNoCert
ldap.user.name.filter=(&(objectclass=person)(mail=%v1))
ldap.user.cert.filter=(&(objectclass=person)(cn=%v1))
ldap.group.name.filter=(&(cn=%v1)(|(objectclass=groupofnames)
(objectclass=groupofuniquenames)))
ldap.group.memberAttributes=member uniquemember
ldap.idleConnection.timeout=600
ldap.waitToRetryConnection.interval=300
ldap.search.timeout=10
ldap.cache.timeout=0
ldap.key.fileName= /opt/IBM/HTTPServer/tds-onlysigners.kdb
ldap.key.file.password.stashFile= /opt/IBM/HTTPServer/tds-onlysigners.sth
ldap.group.search.depth=3
```

4. Create a key database.

   Since we use an SSL transport channel, this means that we need to have a keystore, a stash file, and likely, some signer certificates in that keystore.

   a) Open iKeyman and create a new key database file with the same name as the one defined in the LDAP configuration file. Make sure the path is accurate for your environment. For details on using iKeyman, see Using iKeyman to create a key database file.

   b) Define the password and select "Stash the password to a file."

   c) Select the Signer Certificates option in the dropdown list and click Populate.

   d) Select the CA Certificates that you need. Make sure you have the correct signer certificates.

   e) Click OK. The certificates will be added to your database key file. Close iKeyman.

5. Use the **ldapstash** command to create an LDAP stash file, containing the password for the key database file.

   IHS provides a small executable named `ldapstash.exe` that can create the stash file.

   Usage: `ldapstash <password> <file>`

   `ldapstash.exe WebAS /opt/IBM/HTTPServer/tds-onlysigners.sth`

6. Restart IBM HTTP Server. You can restart the server from the WebSphere Application Server administrative console.
7. In the WebSphere Application Server administrative console, select Security > Global Security. Under Application security, make sure that "Enable application security" is not selected. If you make any changes, restart WebSphere Application Server.
8. Access the TRIRIGA SSO URL in your browser. For example, `http://localhost`.
9. You should get prompted for a login and password. Enter your TDS server login and password. You should be authenticated.

V. Configuring SSO without SSL

About this task

If you want to set up your SSO environment without SSL, make the changes below.

Attention: These settings are **not** recommended for a production server. But they can be useful for a testing environment.

Procedure

1. Three lines in the `ldap.prop` file need to be changed for a non-SSL environment as compared to the LDAP configuration with SSL:

```
ldap.URL=ldap://tdsserver.company.com/ou=organization,o=company.com?mail
ldap.group.URL=ldap://tdsserver.company.com/ou=groups,o=company.com
ldap.transport=TCP
```

   a) You need to change the `ldap.URL` from `ldap://tdsserver.company.com:sslport` (636 is the SSL port for LDAP) to `ldap://tdsserver.company.com`.

   b) You also need to change the `ldap.transport` to use TCP instead of SSL.

2. Comment out these properties:

```
ldap.key.fileName=/opt/IBM/HTTPServer/tds-onlysigners.kdb
ldap.key.file.password.stashFile=/opt/IBM/HTTPServer/tds-onlysigners.sth
ldap.application.password.stashFile
```

Configuring SSO for TRIRIGA on traditional WebSphere with IIS and AD

There are several steps for configuring single sign-on (SSO) with traditional WebSphere Application Server, Microsoft Internet Information Services (IIS), and Microsoft Active Directory (AD).

Contents

I. Installing WebSphere Application Server web server plug-in on IIS

Install the WebSphere Application Server web server plug-ins. The plug-ins are available in the Supplements package for WebSphere Application Server on Passport Advantage. As with the WebSphere Application Server installation, you use the IBM Installation Manager to install the web server plug-ins.

For details on obtaining the Supplements package from Passport Advantage, see: Part numbers of WebSphere software used by IBM TRIRIGA (http://www-01.ibm.com/support/docview.wss?uid=swg21692375).

II. Configuring traditional WebSphere property *trustedSensitiveHeaderOrigin*

There was a change on traditional WAS 9.0.0.11 that added a new configuration property named *trustedSensitiveHeaderOrigin*.

See reference: Potential WebSphere Application Server problems when deployed behind a WebSphere-aware proxy server

On traditional WebSphere, the property is configured as an HTTP channel custom property. This property has a default value of "none", which means that a subset of WebSphere-specific HTTP headers will not be trusted from any host. The property also accepts value a of "*" (all), or a comma-separated list of IP addresses. For a secure deployment in which proxy servers are used, the *trustedSensitiveHeaderOrigin* property should be configured with a comma-separated list of IP addresses corresponding to those of any WebSphere-aware proxy servers in front of the WebSphere server.

Alternatively, to enable the original unsecured behavior, set *trustedSensitiveHeaderOrigin="*"*, which will direct the WebSphere server to trust all headers sent from any host or proxy. This value must only be used for testing, or if the WebSphere server is isolated from external connections.

For traditional WAS, set trustedSensitiveHeaderOrigin as a custom property of HTTP channel.

See reference: HTTP transport channel custom properties

III. Configuring IIS to pass web requests to WebSphere Application Server

About this task

If you are using an already installed web server plug-in on the web server, reconfigure it to use the web server plug-in by using the following procedure.

Note: If the web server plug-in is not installed, then install it, but do not use the following procedure since it is completed automatically during web server plug-in installation. You need to complete the steps below only if you are reconfiguring IIS Version 8.x to use an existing web server plug-in.

Procedure

1. On the Server Manager screen, click Tools > Internet Information Services (IIS) Manager. This action starts the IIS application, and creates a new virtual directory for the website instance that you intend to use with WebSphere Application Server. These instructions assume that you are using the default website.
2. Expand the tree until you see Default Web Site.
3. Right-click Default Web Site and select Add Virtual Directory to create the directory with a default installation.
4. On the Virtual Directory Alias window, enter `sePlugins` in the Alias field.
5. In the Physical Path field of the Web Site Content Directory window, browse to and select the `plugins_root\bin\IIS_web_server_name` directory and click OK. For example, select `C:\Program Files\IBM\WebSphere\Plugins\bin\IIS_webserver1`.
6. Click Test Settings. If the settings test fails, then either change the permissions of the physical directory, or select Connect As, and let IIS connect as a Windows user account that has authority to files in that physical path.

   

   Attention: When you click Test Settings, you might encounter the following warning message if you use the default Pass-thru authentication setting: "`Cannot verify access to path`". For more information, see the Microsoft documentation on this subject.
7. Click OK to add the sePlugins virtual directory to your default website.
8. In the navigation tree, select the sePlugins virtual directory that you created.
9. On the Features panel, double-click Handler Mappings, and then click Edit Feature Permissions on the Actions panel.
10. Select Script and Execute, if they are not already selected then click OK.
11. Manually copy the plug-in binaries to the `plugins_root\bin\IIS_web_server_name` directory. For example, copy the plug-in binary files to the `C:\Program Files\IBM\WebSphere\Plugins\bin\IIS_webserver1` directory.

    Note: The `plugin-cfg.loc` file resides in this directory. The first line of the `plugin-cfg.loc` file identifies the location of the `plugin-cfg.xml` file.

12. Return to the IIS Manager window, and expand the Web Sites folder in the left-hand navigation tree of that window.
13. Select Default Web Site in the navigation tree.
14. Add the Internet Services Application Programming Interface (ISAPI) filter into the IIS configuration.
15. On the Default Web Site Properties panel, complete the following steps.
    a) Double-click the ISAPI Filters tab.
    b) Click to open the Add/Edit Filter Properties dialog.
    c) Enter `iisWASPlugin` in the Filter name field.
    d) Click Browse to select the plug-in file that is located in the `plugins_root\bin\IIS_web_server_name\iisWASPlugin_http.dll` directory.
    e) Click OK to close the Add/Edit Filter Properties dialog.
16. In the navigation tree, select the top level server node.
17. On the Features panel, double-click ISAPI and CGI Restrictions, and then, on the Actions panel, click Add.
18. To determine the value to specify for the ISAPI or CGI Path property, browse to and select the same plug-in file that you selected in the previous step. For example, `plugins_root\bin\IIS_web_server_name\iisWASPlugin_http.dll`.
19. Enter `WASPlugin` in the Description field, select Allow extension path to execute, and click OK to close the ISAPI and CGI Restrictions dialog.
20. Set the value in the `plugin-cfg.loc` file to the location of the configuration file at `plugins_root\config\webserver_name\plugin-cfg.xml`.
    - The default location of `plugin-cfg.xml` is `C:\Program Files\IBM\WebSphere\Plugins\config\IIS_webserver1\plugin-cfg.xml`. The location varies depending on how you have configured your system. If the web server and WebSphere Application Server are on separate machines, you have a remote installation.
    - If the web server and WebSphere Application Server are on the same machine, then you have a local installation, and the correct location of the configuration file might be set. For example, `C:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml`.
    - If the two servers are on the same machine, and the application server is federated, you have a local distributed installation. For example, `C:\IBM\WebSphere\AppServer\profiles\custom01\config\cells\dmgrcell\nodes\managed_node\servers\webserver1\plugin-cfg.xml`.

IV. Configuring SSO with Microsoft IIS

About this task

After configuring IIS to pass web requests to WebSphere, the next step is to set up SSO.

Procedure

1. On the application server, set the following attributes in the `TRIRIGAWEB.properties` file. This file should be located in the `Tririga/config` folder.

```
SSO=Y
SSO_REMOTE_USER=Y
SSO_REMOVE_DOMAIN_NAME=Y
SSO_REQUEST_ATTRIBUTE_NAME=sm_user
```

If AD contains usernames with inconsistent cases (for example, if TRIRIGA users have been entered in lower case, and users in AD are in mixed cases), use the following setting to turn off the case-sensitive check upon login:

```
USERNAME_CASE_SENSITIVE=N
```

If you want to force users to log in through SSO, you must prevent them from using the default login page. Provide an alternative login page that does not contain a username, password, or login button. Use the following settings to specify the alternative login page and directory:

```
ALTERNATE_INDEX_HTML=<indexFileName.html>
ALTERNATE_RESOURCE_DIRECTORY=/<pathToTRIRIGA>/userfiles/alt
```

2. Restart the application server to use the new settings.
3. On the IIS server, right-click My Computer and select Manage.
4. Expand Services and Applications. Select and expand Internet Information Services (IIS) Manager.
5. Under IIS, expand Web Sites.
6. Right-click Default Web Site and select Properties.
7. In the Default Web Site Properties panel, select the Directory Security tab. In Authentication and access control, click Edit.
8. In the Authentication Methods panel, uncheck the box next to Enable anonymous access.
9. Select the type of authentication. If you do not know which one to set, do not choose all three. Try checking one at a time, restarting IIS after the change, and testing to see if the setting works correctly. The correct setting depends on your company's security setup.
   - Integrated Windows authentication
   - Digest authentication for Windows domain servers
   - Basic authentication (password is sent in clear text)
10. Enter the domain name in Default Domain and Realm. The fields that are available depend on the check boxes selected in Authenticated access.
11. Click OK.
12. Restart IIS and make sure that the application server has been restarted.
13. Make sure you have a login within TRIRIGA that matches your domain login. If your domain login is john.doe, the user name in the profile of the TRIRIGA Employee record should be john.doe.

    Note: Logins are case-sensitive. Some logins in Active Directory are stored in mixed case.
14. Open your browser to `http://<<webserver>>/`. It should take you directly to TRIRIGA.

Configuring SSO for TRIRIGA on traditional WebSphere with Apache and TDS

There are several steps for configuring single sign-on (SSO) with traditional WebSphere Application Server, Apache HTTP Server, and Tivoli Directory Server (TDS).

Contents

I. Installing WebSphere Application Server web server plug-in

Install the WebSphere Application Server web server plug-ins. The plug-ins are available in the Supplements package for WebSphere Application Server on Passport Advantage. As with the WebSphere Application Server installation, you use the IBM Installation Manager to install the web server plug-ins.

For details on obtaining the Supplements package from Passport Advantage, see: Part numbers of WebSphere software used by IBM TRIRIGA (http://www-01.ibm.com/support/docview.wss?uid=swg21692375).

II. Configuring traditional WebSphere property *trustedSensitiveHeaderOrigin*

There was a change on traditional WAS 9.0.0.11 that added a new configuration property named *trustedSensitiveHeaderOrigin*.

See reference: Potential WebSphere Application Server problems when deployed behind a WebSphere-aware proxy server

On traditional WebSphere, the property is configured as an HTTP channel custom property. This property has a default value of "none", which means that a subset of WebSphere-specific HTTP headers will not be trusted from any host. The property also accepts value a of "*" (all), or a comma-separated list of IP addresses. For a secure deployment in which proxy servers are used, the *trustedSensitiveHeaderOrigin* property should be configured with a comma-separated list of IP addresses corresponding to those of any WebSphere-aware proxy servers in front of the WebSphere server.

Alternatively, to enable the original unsecured behavior, set *trustedSensitiveHeaderOrigin="*"*, which will direct the WebSphere server to trust all headers sent from any host or proxy. This value must only be used for testing, or if the WebSphere server is isolated from external connections.

For traditional WAS, set trustedSensitiveHeaderOrigin as a custom property of HTTP channel.

See reference: HTTP transport channel custom properties

III. Configuring Apache to pass web requests to WebSphere Application Server

About this task

After you install Apache HTTP Server, you configure it to forward requests to the application server.

The following steps demonstrate how to configure the web server by using the WebSphere Customization Toolbox.

Procedure

1. Start the WebSphere Customization Toolbox.
2. In the Web Server Plug-in Configuration box, select Create.
3. In the Web Server Selection window, select Apache Web Server.
4. In the Web Server Architecture Selection window, select the web server architecture: 64-bit or 32-bit.
5. In the Web Server Configuration File Selection dialog, select the Apache Web Server `httpd.conf` file and port. For example, the file location might be `C:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf\httpd.conf`.
6. In the Web Server Definition Name dialog, specify a unique web server definition, such as the default `webserver1`.
7. In the Configuration Scenario Selection window, specify the location of the application server. If your configuration scenario is local, browse to the location of the `\AppServer` folder. For example, a common location for the application server is `C:\Program Files (x86)\IBM\WebSphere\AppServer`.

8. In the WebSphere Application Server Profile Selection window, select the WebSphere Application Server profile to configure with the current web server plug-in. For example, AppSrv01.
9. In the Plug-in Configuration Summary dialog, review the items you chose and select Configure.

IV. Configuring SSO with Apache HTTP Server

About this task

After you configure Apache HTTP Server to forward web requests to WebSphere Application Server, you set up SSO with Apache HTTP Server.

Procedure

1. On the Apache HTTP Server, in the conf.d directory, create a file called ldap.conf.
2. Add the following content to ldap.conf. Change the parameters tdsserver and yourcompany to suit your TDS setup.

```
LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
LoadModule ldap_module modules/mod_ldap.so

LDAPVerifyServerCert off

<Location "/">
   AuthName "TRIRIGA Tivoli Directory Server Apache SSO"
   AuthType Basic
   AuthBasicProvider ldap
   AuthLDAPURL "ldaps://tdsserver/OU=yourcompany,
   AuthzLDAPAuthoritative off
   Require valid-user
</Location>
```

Configuring SAML SSO for TRIRIGA on traditional WebSphere with TAI

There are several steps for configuring single sign-on (SSO) with traditional WebSphere Application Server, Trust Association Interceptor (TAI), and Security Assertion Markup Language (SAML).

Note: IBM TRIRIGA does not support Security Assertion Markup Language (SAML) as a method of authentication for its non-browser clients. Unsupported non-browser clients include the following clients:

- IBM TRIRIGA CAD Integrator/Publisher
- IBM TRIRIGA Connector for BIM
- IBM TRIRIGA Reservation (VSTO) add-in for Microsoft Outlook

Contents

I. Configuring SSO with TRIRIGA

Procedure

On the application server, set the following attributes in the TRIRIGAWEB.properties file. This file should be located in the Tririga/config folder.

```
SSO=Y
SSO_BACKING_SERVER_PORT=-1
SSO_REMOTE_USER=N
```

```
SSO_USER_PRINCIPAL=Y
SSO_REQUEST_ATTRIBUTE_NAME=uid
```

II. Installing and Configuring SimpleSAMLphp

About this task

SimpleSAMLphp is a PHP-written application that deals with authentication. Its main focus is to provide support for SAML as a Service Provider (SP) or an Identity Provider (IdP). In this example, SimpleSAMLphp is the Identity Provider (IdP).

The following steps demonstrate how to install and configure SimpleSAMLphp.

Procedure

1. Download and install the SimpleSAMLphp package from the SimpleSAMLphp website.
   - For prerequisites and details, see SimpleSAMLphp Installation and Configuration.
2. To enable the IdP functionality, edit the `config.php` file. To enable the IdP to sign its SAML assertions, generate the private key and certificate.
   - For detailed instructions, see the SimpleSAMLphp Identity Provider QuickStart.
3. Configure the metadata for your local Identity Provider (IdP) and remote Service Provider (SP).
   - For detailed instructions, see the SimpleSAMLphp Identity Provider QuickStart.

   In the following example from the `saml20-idp-hosted.php` file, replace `hostname` with the name of your server.

```
<?php
$metadata['hostname'] = array(
   /*
    * The hostname for this IdP. This makes it possible to run multiple
    * IdPs from the same configuration. '__DEFAULT__' means that this one
    * should be used by default.
    */
   'host' => 'hostname.company.com',
   /*
    * The private key and certificate to use when signing responses.
    * These are stored in the cert-directory.
    */
   'privatekey' => 'example.org.pem',
   'certificate' => 'example.org.crt',
   /*
    * The authentication source which should be used to authenticate the
    * user. This must match one of the entries in config/authsources.php.
    */
   'auth' => 'example-userpass',
   'userid.attribute' => 'uid',
   'attributes.NameFormat' => 'urn:oasis:names:tc:SAML:2.0:attrname-format:uri',
   // 'authproc' => array(
   // Convert LDAP names to oids.
   // 100 => array('class' => 'core:AttributeMap', 'name2oid'),
   // ),
   );
```

   In the following example from the `saml20-sp-remote.php` file, replace `<tririgaserver>` with the name of your TRIRIGA server.

```
<?php
$metadata['https://<tririgaserver>:9443/samlsps/trisaml'] = array(
   'AssertionConsumerService' => 'https://<tririgaserver>:9443/samlsps/trisaml',
   'simplesaml.nameidattribute' => 'uid',
   // The URN attribute NameFormat for OID attributes.
   'attributes.NameFormat' => 'urn:oasis:names:tc:SAML:2.0:attrname-format:uri',
   'attributeencodings' => array(
   'urn:oid:1.3.6.1.4.1.5923.1.1.1.10' => 'raw',
   ),
   );
```

III. Configuring SAML SSO with WebSphere Application Server

About this task

After you configure TRIRIGA and SimpleSAMLphp, you set up SAML SSO with WebSphere Application Server. In this example, SimpleSAMLphp is the Identity Provider (IdP) and WebSphere Application Server is the Service Provider (SP).

Before you can use the SAML SSO feature, you must install the SAML Assertion Consumer Service (ACS) application and enable SAML Trust Association Interceptor (TAI). You can install the SAML ACS application in two ways: **Administrative Console** or Python script. Likewise, you can enable SAML TAI properties in two ways: **Administrative Console** or **`wsadmin`** command utility.

The following procedure gives an example in the WebSphere Application Server **Administrative Console**.

Procedure

1. Install the SAML ACS application `WebSphereSamlSP.ear` file. Enable the SAML TAI properties.
   - For detailed instructions, see Enabling your system to use the SAML SSO feature.
   - For a list of SAML TAI properties, see SAML SSO TAI custom properties.
2. Configure the target URL to point to the `requestTest.jsp` URL so that you can see the contents of the HTTP header that are being passed. For example, `https://<tririgaserver>:9443/html/en/default/admin/requestTest.jsp`
   a. To set the URL, select **Security** > **Global Security**.
   b. In the Authentication section, select **Web and SIP Security** > **Trust Association**.
   c. In the General Properties section, select the **Enable Trust Association** check box and click **OK**.
   d. In the Additional Properties section, select **Interceptors** and click **New**.
   e. For **Interceptor Class Name**, enter `com.ibm.ws.security.web.saml.ACSTrustAssociationInterceptor`
   f. Under Custom Properties, set the target URL for **`sso_1.sp.targetUrl`**. For example, `https://<tririgaserver>:9443/html/en/default/admin/requestTest.jsp`
3. Configure the SAML realm as a trusted authentication realm.
   a. To set the realm, select **Security** > **Global Security**.
   b. In the Authentication section, select **RMI/IIOP Security** > **CSIv2 Inbound Communications**.
   c. In the Related Items section, select **Trusted Authentication Realms - Inbound**.
   d. Under Realms, select **Add External Realm** to add your `<hostname>` as a trusted realm.
4. Configure the security role mapping to **All Authenticated in Trusted Realms**.

   Note: This mapping is not the default setting. It is not available until the realm is set in the previous step.
   a. To set the mapping, select **Applications** > **Application Types** > **WebSphere Enterprise Applications** > **[SAML ACS Application (WebSphereSamlSP)]**.
   b. In the Detail Properties section, select **Security Role to User/Group Mapping**.
   c. Select **Map Special Subjects** > **All Authenticated in Trusted Realms**.
5. Now, you can log in from the Identity Provider (IdP) with the URL. In this example, the URL is:

   ```
   https://<hostname.company.com>/simplesamlphp/saml2/idp/SSOService.php
   ?spentityid=https://<tririgaserver>:9443/samlsps/trisaml
   ```

   The IdP will authenticate you. After you accept any certificate warnings, you should see the `requestTest.jsp` page.

6. After you see the `requestTest.jsp` page, you can set the target URL for **`sso_1.sp.targetUrl`** to the proper TRIRIGA URL. For example, `https://<tririgaserver>:9443/`

   Note: Any changes like this, and any changes that are related to security, will require a restart of the application server to be implemented.

   Note: In this example, the **`sso_1.sp.useRealm`** and **`sso_1.sp.realmNameRange`** properties are not really necessary, because the name of the realm is being set as the name of the SAML issuer. The additional certificate-related properties are only necessary when you are using unsigned or self-generated SSL certificates.
7. Once again, log in from the Identity Provider (IdP) with the URL. Once again, the URL is:

   ```
   https://<hostname.company.com>/simplesamlphp/saml2/idp/SSOService.php
   ?spentityid=https://<tririgaserver>:9443/samlsps/trisaml
   ```

   Now, you should see the TRIRIGA home portal.
8. Optional: You can enable the SAML TAI tracing by adding `com.ibm.ws.security.web.saml.*=all`
   a. To set the tracing, select **Troubleshooting** > **Logs and Trace** > **server1** > **Change Log Detail Levels**.
   b. In the **Change Log Detail Levels** text box, enter `com.ibm.ws.security.web.saml.*=all`
   c. After tracing is enabled, the `trace.log` file will contain the tracing results.

Configuring SAML SSO for TRIRIGA on WebSphere Liberty with Okta

There are several steps for configuring single sign-on (SSO) with WebSphere Application Server Liberty as the Service Provider (SP), Okta Identity Provider (IdP), and SP-initiated Security Assertion Markup Language (SAML).

Note: The following SSO content was written for TRIRIGA Application Platform 3.7 and above.

Note: IBM TRIRIGA does not support Security Assertion Markup Language (SAML) as a method of authentication for its non-browser clients. Unsupported non-browser clients include the following clients:

- IBM TRIRIGA CAD Integrator/Publisher
- IBM TRIRIGA Connector for BIM
- IBM TRIRIGA Reservation (VSTO) add-in for Microsoft Outlook

Contents

I. Configuring Okta

Okta provides cloud-based software that deals with identity and access management. In this example, Okta is the Identity Provider (IdP).

To use Okta with SAML, you must create a SAML application on Okta, assign Okta users to TRIRIGA, and copy the Okta Identity Provider metadata file to WebSphere Application Server Liberty.

The following steps demonstrate how to configure Okta.

a. Creating SAML Application on Okta

Procedure

1. Sign in to your **Okta** organization as a user with administrative privileges.

Note: For testing purposes, you can create an Okta developer account from the following URL: https://developer.okta.com/signup/

2. After you sign in to Okta, make sure that you are using the **Classic UI**.
3. From the menu bar, select **Applications**. Select **Add Application**. Then select **Create New App**.
4. In the Create a New Application Integration dialog box, select the following settings.
   a. **Platform**: Select or keep **Web** for the platform.
   b. **Sign On Method**: Select **SAML 2.0** for the protocol to sign in your users.
   c. Click **Create**.
5. In the **General Settings** screen, enter the **App Name**. Click **Next**.
6. In the **Configure SAML** screen, under the SAML Settings section, enter the following settings.
   a. **Single Sign on URL**: Enter the value for: `<TRIRIGA base URL>/ibm/saml20/defaultSP/acs`
   b. **Audience URI (SP Entity ID)**: Enter the value for: `<TRIRIGA base URL>/ibm/saml20/defaultSP`
   c. Click **Next**.
7. In the **Feedback** screen, select **This is an internal app that we have created** for the app type.
8. Click **Finish**.

b. Assigning Okta Users to TRIRIGA

After you create a SAML application on Okta, you must assign Okta users to TRIRIGA. Before you configure the SSO, you must create one or more TRIRIGA users and set each TRIRIGA username to the username of each Okta user. Because after SSO is enabled, the only way for users to log into TRIRIGA will be from the Okta sign-in screen.

Procedure

1. Log in to the TRIRIGA main portal.
2. Create one or more TRIRIGA users. Set each TRIRIGA username to the username of each Okta user.

   Important: You must take this step before you configure the SSO, because after SSO is enabled, the only way for users to log into TRIRIGA will be from the Okta sign-in screen.
3. Sign in to your **Okta** organization as a user with administrative privileges.
4. After you sign in to Okta, make sure that you are using the **Classic UI**.
5. From the menu bar, select **Applications**.
6. Return to your SAML application on Okta. Click the **Assignments** tab.
7. Select **Assign** and then select either **Assign to People** or **Assign to Groups**.
8. Enter the people and groups for whom you want to use SSO with your SAML application. For each, click **Assign**.
9. For any people that you assign, verify the user-specific attributes. Click **Save and Go Back**.
10. Click **Done**.

c. Copying Okta Identity Provider Metadata File to WebSphere Liberty

After you create a SAML application on Okta, and assign Okta users to TRIRIGA, you must copy the Okta Identity Provider metadata file to WebSphere Application Server Liberty.

Procedure

1. Sign in to your **Okta** organization as a user with administrative privileges.
2. After you sign in to Okta, make sure that you are using the **Classic UI**.
3. From the menu bar, select **Applications**.

4. Return to your SAML application on Okta. Click the **Sign On** tab.
5. Right-click the **Identity Provider Metadata** link.
6. Select **Save Link As...**. Rename and save the file as: `idpMetadata.xml`
7. Copy the `idpMetadata.xml` file to WebSphere Liberty at: `<path_to_liberty>/wlp/usr/servers/<server_name>/resources/security`

II. Configuring SSO with TRIRIGA

Procedure

On the application server, set the following attributes in the `TRIRIGAWEB.properties` file. This file should be located in the `Tririga/config` folder.

```
SSO=Y
SSO_REMOTE_USER=N
SSO_USER_PRINCIPAL=Y
SSO_SINGLE_SIGN_OUT_REDIRECT_URL=https://<your Okta domain>/login/signout
```

III. Configuring SAML SSO with WebSphere Liberty

About this task

After you configure Okta and TRIRIGA, you set up SAML SSO with WebSphere Application Server Liberty. In this example, Okta is the Identity Provider (IdP) and WebSphere Application Server Liberty is the Service Provider (SP).

You set up SAML SSO by editing the WebSphere Application Server Liberty `server.xml` file.

Procedure

1. On WebSphere Application Server Liberty, in the `server.xml` file, add the following element declaration inside the `<featureManager>` element.

   ```
   <featureManager ... >
      <feature>samlWeb-2.0</feature>
      <feature>transportSecurity-1.0</feature>
   </featureManager>
   ```

2. Add the following element declaration.

   ```
   <samlWebSso20 id="defaultSP"></samlWebSso20>
   ```

3. Map the TRIRIGA role to `ALL_AUTHENTICATED_USERS`.
   - If the `<application-bnd>` element is already defined, then replace it with the following element declaration.
   - If the `<application-bnd>` element is not defined yet, then add the following element declaration inside the `<webApplication>` element.

   ```
   <webApplication ... >
      <application-bnd>
         <security-role name="TRIRIGA_PLATFORM">
            <special-subject type="ALL_AUTHENTICATED_USERS"></special-subject>
         </security-role>
      </application-bnd>
   </webApplication>
   ```

4. If the `<keyStore>` element is not defined yet, then add a default keystore.
   - In this example, the decoded value of "`{xor}Lz4sLCgwLTs=`" is "`password`".

   ```
   <keyStore id="defaultKeyStore" password="{xor}Lz4sLCgwLTs="/>
   ```

5. Set the `invalidateOnUnauthorizedSessionRequestException` attribute to `true`.
   - If the <httpSession> tag is already defined, then verify if it has an attribute named `invalidateOnUnauthorizedSessionRequestException`.
     - If the attribute is already defined, then verify or change its value to `true`.
     - If the attribute is not defined yet, then add the attribute and set its value to `true`.
   - If the <httpSession> tag is not defined yet, then add the following element declaration.

```
<httpSession invalidateOnUnauthorizedSessionRequestException="true"/>
```

6. Add the following element declaration to change the authentication cache timeout.

```
<authCache timeout="2h"/>
```

7. Save your changes to the `server.xml` file.
8. Start or restart WebSphere Application Server Liberty.

Configuring SSO for TRIRIGA on WebSphere Liberty with Azure and OpenID

There are several steps for configuring single sign-on (SSO) with WebSphere Application Server Liberty, Microsoft Azure, and OpenID Connect (OIDC). Effective April 2020, **Office 365** is now renamed **Microsoft 365**.

Note: The following SSO content was written for TRIRIGA Application Platform 3.7 and above.

Contents

I. Registering OpenID with Azure

To use OpenID Connect (OIDC) with TRIRIGA, you must register an OpenID application with Microsoft Azure, and create an OpenID client secret that will be used later to configure WebSphere Application Server Liberty.

a. Creating Microsoft Azure Account

Procedure

1. If you do not already have a Microsoft Azure account, then create one of the following:
   - Microsoft Azure free account
   - Microsoft 365 developer account
2. After you create an account, open the Microsoft Azure portal.
3. Sign in with your admin credentials.

b. Registering OpenID Application with Microsoft Azure

Procedure

1. Launch the Microsoft Azure app registration screen from the following URL:
   - https://portal.azure.com/#blade/Microsoft_AAD_RegisteredApps/ApplicationsListBlade
2. Select **New Registration** to begin the app registration process.

- **Name**: Enter a descriptive name to display to users. This value is generally used to identify the registration in the Microsoft Azure portal.
- **Account Type**: This value may vary depending on your organization's overall use of Microsoft Azure and Microsoft 365. However, for normal TRIRIGA integrations, **Single Tenant** is sufficient.
- **Redirect URL**: Enter the value for: `<tririga_base_url>/oidcclient/redirect/Azure`

3. Select **Register** to display the next registration screen. From this **Overview** screen, several links either require additional configuration or provide information necessary for WebSphere Application Server Liberty `server.xml` configuration later.
   - **Application (Client) ID**: Copy this value and save it for now. Later, in the `server.xml` file, enter this value in the `clientId` attribute of the `openidConnectClient` element. This step will be discussed later.
   - **Endpoints**. See below.
   - **Certificates & Secrets**. See below.
4. **Endpoints**: This screen provides the endpoints for OpenID interaction.
   a. Copy the **OpenID Connect metadata document** value.
   b. Open the **OpenID Connect metadata document** URL in a new browser tab.
   c. In the `.json` document, locate the values for: `issuer`, `token_endpoint`, `jwks_uri`, and `authorization_endpoint`.
   d. Copy these values and save them for now. Later, in the `server.xml` file, enter these values in the respective `issuerIdentifier`, `tokenEndpointUrl`, `jwkEndpointUrl`, and `authorizationEndpointUrl` attributes of the `openidConnectClient` element. This step will be discussed later.
5. **Certificates & Secrets**: This screen provides the client secret for OpenID interaction. To add a client secret:
   a. Select **New Client Secret**.
   b. Add a description for your client secret.
   c. Select a duration.
   d. Select **Add**.
   e. After you save the configuration changes, the right-most **Value** column will contain the client secret value.
   f. Copy this value and save it for now. Later, in the `server.xml` file, enter this value in the `clientSecret` attribute of the `openidConnectClient` element. This step will be discussed later.

II. Configuring SSO with TRIRIGA

After you register an OpenID application with Microsoft Azure, the next step is to set up SSO with TRIRIGA.

a. Assigning Microsoft Users to TRIRIGA

Before you configure the SSO, you must create one or more TRIRIGA users and set each TRIRIGA username to the username of each Microsoft user. Because after SSO is enabled, the only way for users to log into TRIRIGA will be from the Microsoft sign-in screen.

Procedure

1. Log in to the TRIRIGA main portal.
2. Create one or more TRIRIGA users. Set each TRIRIGA username to the username (email) of each Microsoft user.

Important: You must take this step before you configure the SSO, because after SSO is enabled, the only way for users to log into TRIRIGA will be from the Microsoft sign-in screen.

b. Editing `TRIRIGAWEB.properties` File

Procedure

1. On the application server, set the following attributes in the `TRIRIGAWEB.properties` file. This file should be located in the `Tririga/config` folder.

```
SSO=Y
SSO_REMOTE_USER=N
SSO_USER_PRINCIPAL=Y
```

III. Configuring OpenID with WebSphere Liberty

After you register an OpenID application with Microsoft Azure, and set up SSO with TRIRIGA, the next step is to set up OpenID Connect (OIDC) with WebSphere Application Server Liberty.

You set up OpenID by installing the OpenID Connect client feature, importing the signer certificate into the keystore, and editing the WebSphere Application Server Liberty `server.xml` file.

a. Installing OpenID Connect Client Feature

Procedure

1. Connect to the environment where WebSphere Application Server Liberty is installed.
2. Run the following commands from the `<path_to_liberty>/wlp/bin` folder.

```
installUtility install openidConnectClient-1.0
installUtility install transportSecurity-1.0
```

Note: If you encounter an error that the feature is already installed, ignore the error and proceed to the next step.

b. Importing Signer Certificate into Keystore

Procedure

1. Download the Baltimore CyberTrust Root certificate that is used by Microsoft, from the following URL:
   - https://cacert.omniroot.com/bc2025.crt

   Note: Depending on your system, the certificate file might be named `bc2025.crt` or `bc2025.cer`.
2. After you download the certificate file, copy it to WebSphere Liberty at: `<path_to_liberty>/wlp/usr/servers/<server_name>/resources/security` where `<server_name>` is the name of your WebSphere Liberty server.
3. In the `server.xml` file, make the following changes.
   a. If the `<keyStore>` element is not defined yet, then add a default keystore.

```
<keyStore id="defaultKeyStore" password="password"/>
```

   - To load the keystore file, the `password` value can be stored in clear text or encoded form.
   - To encode a different password, use the WebSphere Liberty `securityUtility` command.

   b. If the `transportSecurity-1.0` feature is not enabled yet, then add the following element declaration inside the `<featureManager>` element.

```
<featureManager ... >
   <feature>transportSecurity-1.0</feature>
</featureManager>
```

c. Save your changes to the `server.xml` file.

d. Start or restart WebSphere Application Server Liberty.

4. Connect to the environment where WebSphere Application Server Liberty is installed.
5. Run the following command from the `<path_to_liberty>/wlp/usr/servers/<server_name>/resources/security` folder.

```
keytool -importcert -keystore <server_keystore_name> -storepass <server_keystore_password>
   -alias loginMicrosoft -file <certificate_filename> -noprompt
```

For example:

```
keytool -importcert -keystore key.p12 -storepass password
   -alias loginMicrosoft -file bc2025.cer -noprompt
```

c. Editing WebSphere Liberty `server.xml` File

Procedure

1. On WebSphere Application Server Liberty, in the `server.xml` file, add the following element declaration inside the `<featureManager>` element.

```
<featureManager ... >
   <feature>openidConnectClient-1.0</feature>
</featureManager>
```

2. Add the following element declaration with the values that you saved earlier from the Microsoft Azure app registration.

```
<openidConnectClient
   clientId="<application id from your registered app>"
   clientSecret="<client secret that you created for your app>"
   id="Azure"
   issuerIdentifier="<issuer from OpenID Connect metadata document>"
   tokenEndpointUrl="<token_endpoint from OpenID Connect metadata document>"
   jwkEndpointUrl="<jwks_uri from OpenID Connect metadata document>"
   authorizationEndpointUrl="<authorization_endpoint from OpenID Connect metadata document>"
   signatureAlgorithm="RS256"
   userIdentityToCreateSubject="preferred_username"
   redirectToRPHostAndPort="https://<public host name>:<ssl port>"
   >
</openidConnectClient>
```

3. Map the TRIRIGA role to `ALL_AUTHENTICATED_USERS`.
   - If the `<application-bnd>` element is already defined, then replace it with the following element declaration.
   - If the `<application-bnd>` element is not defined yet, then add the following element declaration inside the `<webApplication>` element.

```
<webApplication ... >
   <application-bnd>
      <security-role name="TRIRIGA_PLATFORM">
         <special-subject type="ALL_AUTHENTICATED_USERS"></special-subject>
      </security-role>
   </application-bnd>
</webApplication>
```

4. Set the `invalidateOnUnauthorizedSessionRequestException` attribute to `true`.
   - If the `<httpSession>` tag is already defined, then verify if it has an attribute named `invalidateOnUnauthorizedSessionRequestException`.
     - If the attribute is already defined, then verify or change its value to `true`.
     - If the attribute is not defined yet, then add the attribute and set its value to `true`.

- If the <httpSession> tag is not defined yet, then add the following element declaration.

```
<httpSession invalidateOnUnauthorizedSessionRequestException="true"/>
```

5. Add the following element declaration to change the authentication cache timeout.

```
<authCache timeout="2h"/>
```

6. Save your changes to the `server.xml` file.
7. Start or restart WebSphere Application Server Liberty.
8. Log in to TRIRIGA. Now, you should see the Microsoft sign-in page.

   Important: Remember that each TRIRIGA username was set to the username (email) of each Microsoft user.

Configuring SSO for TRIRIGA on traditional WebSphere with Azure and OpenID

There are several steps for configuring single sign-on (SSO) with traditional WebSphere Application Server, Microsoft Azure, and OpenID Connect (OIDC). Effective April 2020, **Office 365** is now renamed **Microsoft 365**.

Note: The following SSO content was written for TRIRIGA Application Platform 3.7 and above.

Contents

I. Registering OpenID with Azure

To use OpenID Connect (OIDC) with TRIRIGA, you must register an OpenID application with Microsoft Azure, and create an OpenID client secret that will be used later to configure WebSphere Application Server.

a. Creating Microsoft Azure Account

Procedure

1. If you do not already have a Microsoft Azure account, then create one of the following:
   - Microsoft Azure free account
   - Microsoft 365 developer account
2. After you create an account, open the Microsoft Azure portal.
3. Sign in with your admin credentials.

b. Registering OpenID Application with Microsoft Azure

Procedure

1. Launch the Microsoft Azure app registration screen from the following URL:
   - https://portal.azure.com/#blade/Microsoft_AAD_RegisteredApps/ApplicationsListBlade
2. Select **New Registration** to begin the app registration process.
   - **Name**: Enter a descriptive name to display to users. This value is generally used to identify the registration in the Microsoft Azure portal.

- **Account Type**: This value may vary depending on your organization's overall use of Microsoft Azure and Microsoft 365. However, for normal TRIRIGA integrations, **Single Tenant** is sufficient.
- **Redirect URL**: Enter the value for: `<tririga_base_url>/oidcclient/redirect/Azure`

3. Select **Register** to display the next registration screen. From this **Overview** screen, several links either require additional configuration or provide information necessary for WebSphere Application Server configuration later.
   - **Application (Client) ID**: Copy this value and save it for now. Later, in the `server.xml` file, enter this value in the `clientId` attribute of the `openidConnectClient` element. This step will be discussed later.
   - **Endpoints**. See below.
   - **Certificates & Secrets**. See below.
4. **Endpoints**: This screen provides the endpoints for OpenID interaction.
   a. Copy the **OpenID Connect metadata document** value.
   b. Open the **OpenID Connect metadata document** URL in a new browser tab.
   c. In the `.json` document, locate the values for: `issuer`, `token_endpoint`, `jwks_uri`, and `authorization_endpoint`.
   d. Copy these values and save them for now. Later, in the `server.xml` file, enter these values in the respective `issuerIdentifier`, `tokenEndpointUrl`, `jwkEndpointUrl`, and `authorizationEndpointUrl` attributes of the `openidConnectClient` element. This step will be discussed later.
5. **Certificates & Secrets**: This screen provides the client secret for OpenID interaction. To add a client secret:
   a. Select **New Client Secret**.
   b. Add a description for your client secret.
   c. Select a duration.
   d. Select **Add**.
   e. After you save the configuration changes, the right-most **Value** column will contain the client secret value.
   f. Copy this value and save it for now. Later, in the `server.xml` file, enter this value in the `clientSecret` attribute of the `openidConnectClient` element. This step will be discussed later.

II. Configuring SSO with TRIRIGA

After you register an OpenID application with Microsoft Azure, the next step is to set up SSO with TRIRIGA.

a. Assigning Microsoft Users to TRIRIGA

Before you configure the SSO, you must create one or more TRIRIGA users and set each TRIRIGA username to the username of each Microsoft user. Because after SSO is enabled, the only way for users to log into TRIRIGA will be from the Microsoft sign-in screen.

Procedure

1. Log in to the TRIRIGA main portal.
2. Create one or more TRIRIGA users. Set each TRIRIGA username to the username (email) of each Microsoft user.

   Important: You must take this step before you configure the SSO, because after SSO is enabled, the only way for users to log into TRIRIGA will be from the Microsoft sign-in screen.

b. Editing `TRIRIGAWEB.properties` File

Procedure

1. On the application server, set the following attributes in the `TRIRIGAWEB.properties` file. This file should be located in the `Tririga/config` folder.

```
SSO=Y
SSO_REMOTE_USER=N
SSO_USER_PRINCIPAL=Y
```

III. Configuring OpenID with WebSphere Application Server

After you register an OpenID application with Microsoft Azure, and set up SSO with TRIRIGA, the next step is to set up OpenID Connect (OIDC) with WebSphere Application Server.

You set up OpenID by installing the OpenID Connect application, importing the signer certificate into the truststore, and configuring the WebSphere Application Server settings in the WebSphere Integrated Solutions Console.

a. Opening WebSphere Integrated Solutions Console

Procedure

1. Launch the WebSphere Integrated Solutions Console from the following URL:
   - `http://<internal_server_name>:<was_admin_port>/admin`
   - For example, `http://<tririga_server>:9066/admin`
2. For the **User ID**, enter: `admin`.
3. For the **Password**, enter: `admin`.
4. Click **Log In**.

   Note: If the console does not request a **User ID** and **Password**, then enable the administrative security in the following procedure.

b. Enabling WebSphere Administrative Security

Note: If the console already requests a **User ID** and **Password**, then you can skip the following procedure.

Procedure

1. Launch the WebSphere Integrated Solutions Console.
2. Select **Security** > **Global Security**.
3. Click **Security Configuration Wizard**.
4. Make sure to clear **Enable Application Security**. Click **Next**.
5. Select **Federated Repositories**. Click **Next**.
6. Enter your administrative credentials. Click **Next**.
7. Click **Finish**.
8. Click **Save**.

c. Installing OpenID Connect Application

Procedure

1. Launch the WebSphere Integrated Solutions Console.
2. Select **Servers** > **Server Types** > **WebSphere Application Servers**.
3. The **Application Servers** screen provides a list of server names and related node names.

a. Copy the **Name** and **Node** values and save them for now. For example, the **Name** might be `server2`, and the **Node** might be `ip-10-165-194-11Node03`.

b. Later, when you run the Python script `installOIDCRP.py`, include these values in the `<serverName>` and `<nodeName>` of the command. This step will be discussed later.

4. Connect to the environment where WebSphere Application Server is installed by using **PuTTY** (https://putty.org/).
5. Determine the values for `<app_server_root_folder>` and `<profile_name>`. To find these values:

   a. Open the TRIRIGA Administrator Console.

   b. Select **Java Info**.

   c. In the **Java System Properties** page, locate the **Classpath** field.

   d. Copy the `<app_server_root_folder>` value and save it for now. The `<app_server_root_folder>` is the parent folder of the `profiles` folder. For example, this folder might be `/home/tririga/IBM/WebSphere/AppServer`.

   e. Copy the `<profile_name>` value and save it for now. The `<profile_name>` is the child folder of the `profiles`. For example, this folder might be `AppSrv02`.

6. Navigate to the `<app_server_root_folder>/bin` folder.

   ```
   cd <app_server_root_folder>/bin
   ```

   For example:

   ```
   cd /home/tririga/IBM/WebSphere/AppServer/bin
   ```

7. Install the OpenID Connect application by running the Python script `installOIDCRP.py` with the values that you saved earlier.

   ```
   ./wsadmin.sh -p <profile_name> -f installOIDCRP.py install <nodeName> <serverName>
   ```

   For example:

   ```
   ./wsadmin.sh -p AppSvr02 -f installOIDCRP.py install ip-10-165-194-11-Node03 server2
   ```

d. Configuring Security Domain & Authentication Cache Timeout

Procedure

1. Launch the WebSphere Integrated Solutions Console.
2. Select **Security** > **Security Domains**.
3. Click **New**.
4. For the **Name**, enter: `OIDCAzure`. Click **OK**.
5. Open the **OIDCAzure** security domain.
6. In the Assigned Scopes section:

   a. Expand **Cell** > **Nodes** > **<Node Name>** > **Servers**.

   b. Select the TRIRIGA server.

7. In the Security Attributes section:

   a. Expand **Application Security**.

   b. Select **Customize for this Domain**.

   c. Select **Enable Application Security**.

   d. To change the authentication cache timeout, expand **Authentication Mechanism Attributes**.

   e. Select **Customize for this Domain**.

   f. Click **Authentication Cache Settings** to go to the General Properties section.

8. In the General Properties section:
    a. Change the **Cache Timeout** to 120 minutes.
    b. Click **OK**.
9. Scroll to the bottom of the page. Click **OK**.

e. Configuring OpenID Connect Relying Party

Procedure

1. Launch the WebSphere Integrated Solutions Console.
2. Select **Security** > **Security Domains**.
3. Open the **OIDCAzure** security domain.
4. In the Security Attributes section:
    a. Expand **Trust Association**.
    b. Select **Customize for this Domain**.
    c. Select **Enable Trust Association**.
    d. Click **Interceptors** to go to the Interceptors screen.
5. In the Interceptors screen, click **New**.
6. In the General Properties section:
    a. For **Interceptor Class Name**, enter `com.ibm.ws.security.oidc.client.RelyingParty`
    b. Under Custom Properties, click **New** to add the following properties with the values that you saved earlier from the Microsoft Azure app registration.
    c. Add **provider_1.clientId** with the **Value** of *<**application id** from your registered app>*.
    d. Add **provider_1.clientSecret** with the **Value** of *<**client secret** that you created for your app>*.
    e. Add **provider_1.identifier** with the **Value** of `Azure`.
    f. Add **provider_1.issuerIdentifier** with the **Value** of *<**issuer** from OpenID Connect metadata document>*.
    g. Add **provider_1.tokenEndpointUrl** with the **Value** of *<**token_endpoint** from OpenID Connect metadata document>*.
    h. Add **provider_1.jwkEndpointUrl** with the **Value** of *<**jwks_uri** from OpenID Connect metadata document>*.
    i. Add **provider_1.authorizeEndpointUrl** with the **Value** of *<**authorization_endpoint** from OpenID Connect metadata document>*.
    j. Add **provider_1.signatureAlgorithm** with the **Value** of RS256.
    k. Add **provider_1.userIdentifier** with the **Value** of `preferred_username`.
    l. Add **provider_1.redirectToRPHostAndPort** with the **Value** of `https://`*<public host name>*`:`*<ssl port>*.
    m. Click **OK**.
7. Return to the **OIDCAzure** security domain.
8. In the Security Attributes section:
    a. Scroll to the bottom of the page.
    b. Click **Custom Properties** to go to the Custom Properties screen.
9. In the Custom Properties screen, click **New**.
    a. Add **com.ibm.websphere.security.InvokeTAIbeforeSSO** with the **Value** of `com.ibm.ws.security.oidc.client.RelyingParty`.

b. Click **OK**.

f. Configuring Microsoft Azure Realm as Trusted Realm

Procedure

1. Launch the WebSphere Integrated Solutions Console.
2. Select **Security** > **Security Domains**.
3. Open the **OIDCAzure** security domain.
4. In the Security Attributes section, expand **User Realm**.
5. Click **Configure** to go to the next screen.
6. Select **Trusted Authentication Realms - Inbound** to go to the next screen.
7. In the Trust section, select **Trust Realms as Indicated Below**.
8. Under Realms, click **Add External Realm**.
   a. Add the **Name** of <***issuer*** *from OpenID Connect metadata document*>.
   b. Click **OK**.

g. Configuring WebSphere Session Property

Procedure

1. Launch the WebSphere Integrated Solutions Console.
2. Select **Servers** > **Server Types** > **WebSphere Application Servers**.
3. Select the application server. For example, the **Name** might be `server2`.
4. In the Container Settings section, select **Session Management**.
5. In the Additional Properties section, click **Custom Properties** to go to the Custom Properties screen.
6. In the Custom Properties screen, click **New**.
   a. Add **`invalidateOnUnauthorizedSessionRequestException`** with the **Value** of `true`.
   b. Click **OK**.

h. Importing Signer Certificate into Truststore

Procedure

1. Download the Baltimore CyberTrust Root certificate that is used by Microsoft, from the following URL:
   - https://cacert.omniroot.com/bc2025.crt

   Note: Depending on your system, the certificate file might be named `bc2025.crt` or `bc2025.cer`.
2. After you download the certificate file, copy it to the environment where TRIRIGA is installed.
3. Launch the WebSphere Integrated Solutions Console.
4. Select **Security** > **SSL Certificate and Key Management**.
5. In the Related Items section, select **Key Stores and Certificates**.
6. In the Key Stores and Certificates screen, click **NodeDefaultTrustStore**.
7. In the Additional Properties section, select **Signer Certificates**.
8. In the Signer Certificates screen, click **Add**.
   a. Add the **Alias** of `baltimore` with the **File Name** of <*complete path to the* ***bc2025*** *file on the environment where TRIRIGA is installed*>.
   b. Click **OK**.

i. Mapping TRIRIGA Role to All Authenticated Users

Procedure

1. Launch the WebSphere Integrated Solutions Console.
2. Select **Applications** > **Application Types** > **WebSphere Enterprise Applications**.
3. Select the application name that begins with `IBM-TRIRIGA`. For example, the **Name** might be `IBM-TRIRIGA_Build-296099`.
4. In the Detail Properties section, select **Security Role to User/Group Mapping**.
5. In the Security Role to User/Group Mapping screen:
   a. Select the **Role** of `TRIRIGA_PLATFORM`.
   b. Select **Map Special Subjects** > **All Authenticated in Trusted Realms**.
   c. Click **OK**.

j. Saving All Changes to Master Repository

Procedure

1. Launch the WebSphere Integrated Solutions Console.
2. Select **System Administration** > **Save Changes to Master Repository**.
3. Click **Save**.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work
must include a copyright notice as follows:
© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux® is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at https://www.ibm.com/privacy/details/us/en/ in the section entitled “Cookies, Web Beacons and Other Technologies.”

Part Number:

(1P) P/N: