Configuring IBM WebSphere DataPower as the OAuth authorization server

Overview

The MobileFirst security framework is built around an authorization server that implements the OAuth protocol, and exposes the OAuth endpoints with which the client interacts. MobileFirst Server implements custom security logic and advanced security features on top of the authorization server. By default, MobileFirst Server functions also as the OAuth authorization server. However, you can configure IBM® WebSphere® DataPower® (DataPower) to act as the authorization server, and interact with MobileFirst Server. This design provides you with enhanced flexibility in setting up production topologies, for example, deploying the DataPower authorization server in the DMZ.

> **Note:** The basic building blocks of the security framework (security checks and challenge handlers) are unaffected by this mode. The behavior of the building blocks is the same regardless of whether the authorization server is MobileFirst Server or DataPower.

The integration of the MobileFirst security framework with DataPower as the authorization server is achieved by using the provided MobileFirst DataPower pattern file, dp-external-az-pattern.zip. You can get this file from the IBM MobileFirst Operations Console: from the console's **Dashboard**, select **Download Center**, and then select the **Tools** tab. In the **MobileFirst External Authorization Server Pattern** section of the Tools tab, select **Download** and save the pattern to your preferred location.

To use DataPower as the authorization server, deploy the provided pattern to your DataPower appliance and configure MobileFirst Server to interact with DataPower as the authorization server, as outlined in the following procedure.

> Note: When using DataPower as the authorization server, configure client applications to connect to the DataPower appliance instead of connecting directly to MobileFirst Server. For example, in an iOS application, set the **wlServerHost** and **wlServerPort** properties in **mfpclient.plist** to the host IP address and port of the DataPower appliance. If you are using a self-signed SSL certificate for DataPower, you also need to import this certificate into the client application.

Setup

Preliminary steps

1. Create a key pair for the DataPower SSL certificate with a public key named **azserver-sscert.pem** and a private key named **azserver-privkey.pe**m. The exact procedure for creating the SSL key pair for production depends on your certificate authority, and therefore cannot be documented here. However, during development you can run the following commands from a command-line terminal to create a self-signed certificate:

   ```
   openssl req -x509 -newkey rsa:4096 -keyout azserver-privkey.pem -out azserver-sscert.pem -days
   365 -nodes
   ```

2. Create a key to be used by DataPower for encryption of OAuth tokens. The key is a hexadecimal

string whose size is at least 32 bytes. Save the key to a file named key. Following is a sample key: **0xabcd1234abcd1234abcd1234abcd1234abcd1234abcd1234abcd1234abcd1234**.

3. Create a file that is named **secret**. This secret is used by DataPower to authenticate itself with MobileFirst Server. The secret file is a text file that contains a JSON object with a **secret** entry whose string value defines the secret. The following sample defines a "12345" secret:

   ```
   {"secret":"12345"}
   ```

Deploying the MobileFirst DataPower pattern

In the WebGUI of your DataPower appliance,

1. Create a new DataPower domain. **2. Switch to the new domain, and upload the file that contains the public and private SSL-certification keys that you created in preliminary Step 1 (**azserver-sscert.pem** and azserver-privkey.pem**) to the **cert:**/// directory.
2. Upload the **key** file that you created in preliminary Step 2 to the same directory.
3. Upload the **secret** file that you created in preliminary Step 3 to the **local:**/// directory.
4. Select **Blueprint Console** in the WebGUI navigation sidebar. Then select the **Patterns** tab, and import the pattern by selecting the graphical import button (next to the **New Pattern** button): Graphic of the import button..
5. Select the pattern archive file (dp-external-az-pattern.zip), and wait for the import to complete.
6. Select **Deploy**, and provide the input in the four required fields: enter a name for the service, the URL of your MobileFirst Server, and the DataPower IP address and port that you want the authorization server to listen to.
7. After the pattern is deployed, select the **Services** tab to ensure that the new authorization service started successfully.

Configuring MobileFirst Server to work with DataPower

1. Add the following JNDI entries to the application-server configuration file of your MobileFirst Server (**server.xml**). Replace the **your_secret** placeholder with the secret that you defined in your **secret** file in preliminary Step 3 (for example, "12345"). This secret is used for authenticating the DataPower appliance.

   ```
   <jndiEntry jndiName="mfp/mfp.authorization.server" value="external"/>
   <jndiEntry jndiName="mfp.external.authorization.server.secret" value="your_secret"/>
   <jndiEntry jndiName="mfp.external.authorization.server.introspection.url"
      value="https://<DataPower host address>:8443/az/v1/introspection"/>
   ```

2. Import the DataPower public-key certificate file (**azserver-sscert.pem**) into the WebSphere Application Server Liberty keystore of your MobileFirst Server to allow the server to connect over SSL. You can run the following commands to import the key into the keystore:

   ```
   openssl x509 -outform der -in azserver-sscert.pem -out azserver-sscert.der
   keytool -import -keystore key.jks -file azserver-sscert.der
   ```

The first command takes the certificate and converts it into DER format. The second command uses the Java **keytool** utility to import the certificate into the WebSphere Application Server Liberty keystore of your MobileFirst Server.

3. Create a special empty scope-element mapping to be used by DataPower. You need to create this mapping in every application that is registered with MobileFirst Server. To create a mapping, select the **Security** tab on the application page, and then select **Create New** under **Security-Elements Mapping**. In the Add New Scope-Element Mapping dialog window, create a new mapping (for example, a mapping named none), and leave the **Scope Element** field empty (do not map the scope to any security check). Select Add to complete the mapping.

Protecting resources with an empty scope

To protect a resource by requiring an access token, you need to explicitly set the resource's protecting scope to the empty scope element that you mapped in Step 3 of the server-configuration procedure. For information about how to protect your resources with a scope, see OAuth resource protection (../#protecting-resources). On the client side, call WLResourceRequest with a scope parameter that is set to the same empty scope element.