Protecting mobile applications with the ISAM module for IBM DataPower Gateway, Part 1: Securing and optimizing mobile workloads using mobile patterns

Ozair Sheikh (ozairs@ca.ibm.com)
Senior Product Manager
IBM

June 2015
(First published 22 May 2015)

The IBM® Security Access Manager module for IBM DataPower® Gateway delivers strong authentication capabilities to protect mobile applications with multi-factor authentication based on contextual data and enforcement using one-time passwords. Enterprises must protect both consumer and employee mobile applications from malicious attackers to avoid data exposures and unauthorized access to mobile applications. Stronger security can be enforced using multiple authentication factors, often based on "something you know", such as a password, and "something you have", such as a mobile device. In this tutorial, you will learn how to use the ISAM module multi-factor security framework to protect applications based on user credentials.

Introduction

This tutorial describes how you can configure the IBM Security Access Manager (ISAM) module on IBM DataPower Gateway (hereafter called DataPower) to secure mobile applications with form-based authentication and to provide single sign-on (SSO) to backend REST and JSON services. You will also enrich the mobile gateway flow with additional security and integration capabilities using a DataPower Multi-Protocol Gateway service.

In a subsequent tutorial, you will add multi-factor authentication using one-time passwords (OTP) to the reverse proxy deployed on DataPower using the ISAM for Mobile policy authoring engine.

Prerequisites

You will need the following products to complete this tutorial:

- IBM DataPower Gateway V7.1 or above
- ISAM for Mobile V8.0.0.5 or above
- IBM MobileFirst® V6 or V7
This tutorial is based on the **ISAM and MobileFirst integration**, which describes how to protect mobile applications using the ISAM for Mobile appliance and how to provide SSO to the IBM MobileFirst platform.

**Note:** You can apply the capabilities discussed in this tutorial to any backend service provider. You will simply need to change the access control list (ACL) of the protected resource and the SSO mechanism for both the backend service provider in the policy server and reverse proxy configuration.

You will learn how to quickly configure the ISAM module on DataPower in three easy steps:

1. Deploy the reverse proxy pattern.
2. Configure the Policy Server ACL and objects.
3. Enhance the reverse proxy with the DataPower functionality.

The high-level architecture shows the request flow in Figure 1.

**Figure 1. High-level architecture**

![High-level architecture diagram](image)

**Embedded LDAP server**

The reverse proxy (RP) connects over the secure socket layer (SSL) to the embedded LDAP server in the ISAM appliance. The RP requires the KDB and STH files for the LDAP server on DataPower. For simplicity, you can download the entire keystore (private and public keys) and upload them to the DataPower appliance (not recommended for real-world environments).

In the ISAM appliance, select **Manage System Settings > Secure Settings > SSL Certificates** and select the **embedded_ldap_keys** keystore. Select **Manage > Export** and save it to a location on your desktop. Unzip this file to view the crypto files.

1. The mobile application sends a request to the reverse proxy on the DataPower appliance, which then checks the request for a valid session. Otherwise, it prompts the user for a user ID and password.
2. For an unauthenticated session, the user submits their credentials to the reverse proxy on the DataPower appliance, which sends the credentials and resource to the policy server and LDAP server on the ISAM for Mobile appliance for authentication and authorization.
3. After successful authentication and authorization, the reverse proxy generates an HTTP header asserting the identity and passing it to the MobileFirst server, which also validates the identity before allowing access to the protected backend resource.
4. Any subsequent requests does not prompt for authentication since the user session is valid.
This tutorial assumes that you have already deployed both ISAM for Mobile and DataPower and configured IP addresses for both the management and application interfaces.

This tutorial will reference the following two ISAM for Mobile interfaces:

- **Management**: M.1
- **Application**: P.1

You can view the IP address on the ISAM web console home page as shown in Figure 2.

**Figure 2. ISAM for Mobile interfaces**

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>MAC Address</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
<td>M.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td>P.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Important**

The following instructions use the local policy server and embedded Lightweight Directory Access Protocol (LDAP) server in the ISAM for Mobile appliance. You should *not* use the embedded LDAP server in a production environment. It is shown below for development purposes and ease of testing.

Before you execute the following instructions, make sure you have the following information:

- ISAM Policy server host name, port, and password
- LDAP server host name, port (preferably 636), password, and a copy of the LDAP KDB keystore
- DataPower Ethernet Interface(s) IP address. You only need a single IP address to get the configuration to work, but multiple IP addresses (management and application) are recommended. You can access the web graphical user interface (GUI) from the IP address (default port is 9090).

**Step 1: Deploy the reverse proxy pattern**

In this section, you will deploy the reverse proxy service using the DataPower Blueprints console, which will accelerate development time with the deployment of the DataPower services with pre-configured best practices.

The easiest way to configure DataPower objects is to use the vertical navigation bar and enter the object name and appropriate information. The following sections will use this approach to create a DataPower configuration.

**Configure the environment**

1. Login to the **default** domain of the DataPower web GUI and use the vertical navigation bar to find the application domain object (Figure 3) and create a domain called **isam**.
2. Create a **host alias** object called **lab-idg** using the vertical navigation bar. This configuration allows you to externalize the IP address from the DataPower configuration, which aids in easier migration between environments.

3. View the pre-shipped DataPower patterns by opening the **Blueprint Console** link (left hand corner of the GUI). Select the **Patterns** tab and then the **Web application and secure reverse proxy** pattern (Figure 4). Click the **Export** button to save the pattern to your desktop.

4. Repeat the previous step to export the **Mobile Gateway** and **Mobile REST Proxy** patterns.

5. Go back to the web GUI and switch to the **isam** domain. You will perform the remaining steps in this domain.

### Set up the Access Manager runtime

1. Configure the **Access Manager Runtime** object to connect to a policy server and LDAP server. In the vertical navigation bar, enter and select **Access Manager Runtime**. In the **Main** tab, check **enabled** for the **Administrative state**.

2. Select the **Policy Server** tab and enter the IP address (or host name) of your ISAM appliance (ISAM application interface M.1) and leave the default port (7135) and Management Domain (Default).

3. Select the **LDAP** tab and enter the IP address (or host name) of your ISAM appliance (ISAM application interface M.1) and Port 389. You will later override the port number in the reverse proxy configuration. Click **Apply** to save this object.

4. In the vertical navigation bar, enter and select **Password Map** and click the link.

5. Create an alias, **sec_master**, and enter the password for this account. Click **Apply** to save this object.
Deploy the Access Manager reverse proxy

1. Go back to the Blueprint Console. You should be in the isam domain (top right-hand corner shows the domain).
2. Click the Patterns tab and select the Import button as shown in Figure 5.

Figure 5. Import pattern

3. Browse to the location of the Web application and secure reverse proxy pattern you downloaded in the previous step and click Import. Repeat this step to import the other patterns that were previously downloaded.
4. Select the Web application and secure reverse proxy pattern and click Deploy.
5. In the pattern wizard, enter the following information as shown in Figure 6:
   a. Step 1: Service name: dp-reverse-proxy
   b. Step 2: Enter the administrator name of sec_master and administrator password alias of sec_master (you created this object in the last section). Enter the Host value of lab-idg and the Listening Port value (use the default value unless you have a port conflict).
c. **Step 3:** For the front-side connection details, check the front-side HTTP and HTTPS ports. You may need to change these port numbers if another service is using these ports. Select the SSL Certificate Key File, `isamcert://keytab/pdsrv.kdb`, as shown in Figure 7. You should *not* use the HTTP protocol if you are sending secure information, such as a user ID and password. It is selected in this tutorial for ease of testing.
Figure 7. Reverse proxy pattern - client connection

Step 3: Access Manager Reverse Proxy client connection and SSL
Specify whether HTTP and/or HTTPS access is allowed to the Access Manager Reverse Proxy.

- HTTP: [ ]
- HTTP Port: [7080]
- HTTPS: [ ]
- HTTPS Port: [7443]
- SSL Certificate Key File: [isamcert://keytab]
- pdenv.kdb

Step 4: Access Manager Reverse Proxy client authentication methods
Specify the Basic authentication or Forms transport options.

- Basic Authentication Transport: [None]

Step 4: Authentication – Enforce security with forms or HTTP basic authentication. Leave the default Forms option.

Step 5: Backend routing and SSO - Enter the URI (context-root) of the backend application that you are protecting. For this tutorial, it is /worklight as shown in Figure 8. Enter the backend host name and port number of your backend application. Specify whether you want to propagate the authenticated identity using an HTTP header or LTPA token. Check the IV-USER checkbox.

f. Click Deploy Pattern once you have entered all the appropriate values.
6. Once deployed, click the **Services** tab to make sure that the Access Manager reverse proxy is deployed successfully (Figure 9).

**Figure 8. Reverse proxy pattern – security settings**

- **Forms options**
  - Forms Authentication Transport: Both

- **Step 5: Access Manager Junction**
  - Junction Point Name: /worklight

- **Target back-end server**
  - Target Back-end Servers:
  - Host: [Name]
  - TCP or SSL Port: 9080

- **Supply identity information in HTTP headers**
  - **HTTP Header Identity Information:**
    - IV-USER
    - IV-USER-L
    - IV-GROUPS
    - IV-CREDS

**Figure 9. Pattern deployment status**

<table>
<thead>
<tr>
<th>Service</th>
<th>Status</th>
<th>Service Type</th>
<th>Front side URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>dp-reverse-proxy</td>
<td>Up</td>
<td>Access Manager Reverse</td>
<td>http, https</td>
</tr>
</tbody>
</table>

**Customize the Access Manager reverse proxy**

1. In the **Services** tab, click the recently created **dp-reverse-proxy**. This action opens the object configuration in the web GUI.

2. The user registry is currently configured to connect to the LDAP server using port 389, which is not secure. This was done as part of the Access Manager Runtime object configuration. You will now modify the user registry settings to connect using the SSL using port 636. Select the **User Registry** tab and check **Enable SSL**. Upload the LDAP crypto files (**embedded_ldap_keys.kdb** and **embedded_ldap_keys.sth**) to the appliance **isamcert://keytab** directory (you should have a ZIP copy of these files on your workstation). Once uploaded, select the **embedded_ldap_keys.kdb** file as shown in Figure 10.
Figure 10. Reverse proxy user registry

Enable SSL

Port

636

SSL Settings

Key File Name

isamcert://keytab

embedded_ldap_keys.kdb

Certificate Label

3. Select the Junction tab and edit the dp-reverse-proxy_backend junction. Click the Identity tab and change the HTTP Basic Authentication Header value to Ignore as shown in Figure 11. Click Apply to save the configuration of all the modified objects.

Figure 11. Reverse proxy single sign-on

Supply identity information in HTTP headers

HTTP Basic Authentication Header

Ignore

HTTP Header Identity Information

IV-USER

IV-USER-L

IV-GROUPS

IV-CREDS

Note: This setting is required when sending an HTTP header to the IBM MobileFirst server.

Step 2: Set up the Policy Server ACL and objects

1. Open the IBM Security Access Manager (ISAM) console, https://<M.1>, and select Secure Web Settings > Manage > Policy Administration. Login to the policy server using the sec_master account.

2. Expand Object Space > Browse Object Space and additional objects until you see the reverse proxy and junction name as shown in Figure 12. This step validates the connectivity between DataPower and the ISAM policy server. In the next step, you will attach an ACL to the junction worklight to secure the protected resources. If your backend resource is protected using a different junction name, you will see the appropriate name in the policy server object page.
3. Login to the DataPower command line interface (CLI) interface so you can configure ACL on the ISAM objects using the pdadmin commands.

4. Switch to the isam domain (the domain where the Access Manager Runtime object is configured) and enter the following commands shown in Listing 1.

**Listing 1. Login to pdadmin**

```
switch domain isam;  
cd; pdadmin;  
login –a sec_master –p passw0rd
```

5. Enter the following commands shown in Listing 2 to create the unauth, noaccess, and anyauth ACL objects. You only need to create these objects once in the policy server object space.

**Listing 2. Create ACL**

```
#Create ACL for unauthenticated access
acl create unauth  
acl modify unauth set group iv-admin TcmdbsvaBRrxl  
acl modify unauth set group webseal-servers Tgmdbsrxl  
acl modify unauth set user sec_master TcmdbsvaBRrxl  
acl modify unauth set any-other Trx  
acl modify unauth set unauthenticated Trx

#Create ACL to prevent all access
acl create noaccess  
acl modify noaccess set group iv-admin TcmdbsvaBRrxl  
acl modify noaccess set group webseal-servers Tgmdbsrxl  
acl modify noaccess set user sec_master TcmdbsvaBRrxl  
acl modify noaccess set any-other T  
acl modify noaccess set unauthenticated T

#Create ACL allowing any authenticated user to access  
#Copy of default-webseal ACL
acl create anyauth  
acl modify anyauth set group iv-admin TcmdbsvaBRrxl  
acl modify anyauth set group webseal-servers Tgmdbsrxl  
acl modify anyauth set user sec_master TcmdbsvaBRrxl  
acl modify anyauth set any-other Trx  
acl modify anyauth set unauthenticated T
exit
```

You can also download a snippet of these commands in the Download section of the tutorial.
6. Attach the **unauth** ACL to the root junction object. This rule allows access to the top-level root folder. You will define fine-grained rules to manage access to secure resources at a lower level in the next step.

   ```
   acl attach /WebSEAL/lab-idg-reverse-proxy/worklight unauth
   ```

7. Attach the **anyauth** ACL to an object below the root junction object containing a list of protected resources. This URI is used to call secure resources on the MobileFirst server, which must contain a valid HTTP header.

   ```
   acl attach /WebSEAL/lab-idg-reverse-proxy/worklight/apps/services/api/anyauth
   ```

   **Note:** This step reflects the URI pattern of the protected resources. You can change this URI pattern (`/worklight/apps/services/api/`) to a different string for your application-specific protected resources.

8. Create a non-admin user **demo** for testing:

   ```
   user create -no-password-policy demo cn=demo,cn=SecurityDaemons,secAuthority=Default demo demo demo
   user modify demo account-valid yes
   ```

9. Click **Exit** once you are finished and log out of the CLI prompt. You are now ready to test the reverse proxy policy.

   You can run a simple test of the Access Manager reverse proxy configuration using the web browser.

10. Open a web browser and enter the following URL: `https://<lab-idg>:7443/worklight` (enter a different port number if you used a different value). You are prompted to enter a user name and password. Enter the user name of **demo** and password of **demo** (you created this earlier on the command line). Upon success, you are redirected to the home page as shown in Figure 13.

   **Figure 13. Test reverse proxy**

   ![ISAM for DataPower welcome demo](image)

   Realm ID: HeaderAuthRealm
   User ID: demo

---

**Step 3: Enhance the reverse proxy with a DataPower policy**

In this step, you will deploy a Mobile REST proxy pattern with different rules to secure and optimize mobile workloads based on the HTTP method. This service will later be combined with the existing Access Manager reverse proxy object.

1. Open the Blueprint Console and select the **Patterns** tab. Select the **Mobile REST proxy** pattern and click **Deploy** (you have previously imported this pattern).
2. In the pattern wizard, enter the following information that is also shown in Figure 14:
a. **Service name**: rest-proxy
b. **URL**: http://<hostname>:9080/ (the host name is the IBM MobileFirst server)
c. **IP address**: 127.0.0.1
d. **Port**: 8000

**Figure 14. Deploy REST proxy pattern**

**Deploy pattern**

Mobile REST proxy

- **Service name**: dp-rest-proxy

**Step 2: Back-end endpoint details**

Specify the URL of the back-end server for which DataPower acts as a web proxy. Also, optionally, provide an existing SSL proxy.

- **URL**: http://<hostname>:9080/
- **SSL proxy profile**

**Step 3: Front-end endpoint details**

Specify the IP address and port number of the HTTP front-side handler (FSH).

- **IP address**: 127.0.0.1
- **Port**: 8000

3. Click **Deploy Pattern**. In the **Services** tab, click the **rest-proxy** pattern and open the **rest-proxy** processing policy.

4. This policy defines different rules for the various HTTP verbs (GET, PUT, POST, and DELETE) as shown in Figure 15. For example, the HTTP POST rule defines the following:
   a. **Match**: This matches the HTTP POST requests.
   b. **Convert Query Params to XML**: This converts the input parameter into an XML node for processing by any actions that require XML.
   c. **SLM**: This enforces the service level agreements (SLA) and traffic control policies.
   d. **GatewayScript**: This secures and optimizes the JSON or binary payload. For example, it performs off-box calls using "url-open" asynchronously with the JSON input.
   e. **Result**: This copies the results of the previous actions and sends them to the backend service.
You will now connect the existing reverse proxy service, `dp-reverse-proxy`, with the recently deployed `dp-rest-proxy` Multi-Protocol Gateway service.

1. Open the Access Manager reverse proxy object named `dp-reverse-proxy`. Select the Junction tab and edit the `dp-reverse-proxy_backend` junction.
2. Click the Server tab and change the host field to `127.0.0.1` and TCP or SSL port field to `7000` (or the matching port of the dp-rest-proxy). This change sends all traffic to the `dp-rest-proxy` Multi-Protocol Gateway service (also listening on interface `127.0.0.1` and the same port number), which then interacts with the backend mobile application (Figure 16).

Figure 16. Junction backend

3. Test the web application again and remove any session cookies from the previous tests. Open a web browser and enter the following URL: `https://<lab-idg>:7443/worklight`. You are prompted to enter a user name and password. Enter the user name of `demo` and password of `demo` (you created this earlier on the command line). Upon success, you are redirected to the home page.
4. You can enable the probe in the dp-rest-proxy service to view the request and response messages. The current set of rules do not modify the messages, but you can apply a GatewayScript action to enrich the message, or a schema to validate any JSON payloads, or even enforce service-level-agreements (SLA) using the SLM policy.

Conclusion

In this tutorial, you learned how to deploy a reverse proxy service using the Blueprints console on the IBM DataPower Gateway appliance. The reverse proxy was then configured to secure protected resources and provide single sign-on to the IBM MobileFirst platform. Finally, you deployed a REST gateway pattern using the Blueprint console to secure and optimize the reverse proxy workload.

In Part 2, you will extend the security policy to include enforcement of a one-time password. This policy will require an OTP authentication level step-up when a "high value" transaction is attempted. The enforcement of this policy will be done using the ISAM reverse proxy on DataPower. The policy definition and runtime evaluation will be done using ISAM for Mobile™.
Acknowledgments

The author would like to thank members of the ISAM and DataPower Engineering teams for their review of this tutorial: **David Moore, Jon Palgon, Davin Holmes, Scott Andrews, and Leo Farrell.**
## Downloads

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code sample</td>
<td>code_sample.zip</td>
<td>1KB</td>
</tr>
</tbody>
</table>
Resources

- WebSphere DataPower Knowledge Center
- IBM Security Access Manager
- developerWorks WebSphere zone
- developerWorks Security resource page
- WebSphere DataPower discussion forum
About the author

Ozair Sheikh

Ozair Sheikh is a Senior Product Line Manager for IBM DataPower Gateway and certified IBM IT Specialist. He is an experienced technical and business product leader with over 10 years in managing, consulting, instructing, and developing enterprise solutions. He is an avid speaker at several worldwide conferences on topics such as mobile security, API Management, and building mission-critical gateway architectures. In his current role, Ozair helps drive new innovative solutions for the DataPower Gateway platform that reflect customer requirements and market trends.

© Copyright IBM Corporation 2015
Trademark
(www.ibm.com/developerworks/ibm/trademarks/)