Lab 14: Web Services Security

In this lab, you will experiment with Web services security.

Objectives:

- To demonstrate the WS-Security specification

Part 1: Creating the Service

In this part, you will create a service and client that you will later secure.

Steps:

_1. Create a new Dynamic Web project named **lab14Web** associated with *lab14EAR*.

_2. As you have done in a previous lab, write a service endpoint class with named **HelloWorld** in the *hello* package with a single method that accepts a String and returns a String. You can copy code from the "Hello, World" lab you did earlier.

_3. Next, create another Dynamic Web Project for the client named **lab14ClientWeb** associated with *lab14ClientEAR*.

_4. Generate client stub classes for the service into the new project.

_5. Create a servlet named **HelloServlet** in a new package named *servlets* that invokes the service from its doGet method. You can use dependency injection to obtain the service reference. See Lab 7 as a review.

_6. Test the unsecured service by running the servlet on the server.

_7. Ensure that the TCP/IP monitor is set up, then modify the servlet as necessary so it sends messages to the monitor. Lab 7 has some hints.

Test to ensure that the monitoring works.

Part 2: Securing the Service

In this part, you will configure simple security that could be used in production, though you will not take the process to the ultimate conclusion.

Steps:

_1. In this step, you configure a simple security policy on the application server that will require a username/password token on the service. Follow this procedure:

a. In the RAD Servers tab, right-click on the WebSphere server and choose *Administration - Run Administrative Console*.

b. In the left-hand navigation tree, expand the *Services - Policy Sets* entry then click on *General client policy set bindings*.

c. In the bindings list, click on *Client sample*.

d. In the Client sample bindings, click on *WS-Security*.

e. In the WS-Security settings, click on *Authentication and protection*.

Copyright © Descriptor Systems 2014. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

f. In the Authentication tokens list, click on *gen_signunametoken*.

g. In the Additional Bindings section, click on *Callback handler*.

h. Under Basic Authentication, enter a user name and password of your choice, for example **sam** and **1234**.

Press Apply.

i. At the top of the page, click the Save link, then logout of, and exit the administrative console.

_2. Next, configure the Web service so it requires a username token:

a. In the service project, *lab14Web*, in the Enterprise Explorer, expand the Services folder.

Right-click on the *{http://hello}HelloWorldService* entry and choose *Manage Policy Set Attachments - Server Side* to start the wizard.

b. Press the Add button, then choose:

```
Endpoint:        all endpoints
Policy set:      Username WSSecurity default
General binding: Provider sample
```

Press OK, followed by Ignore and Finish.

_3. Next, configure the client project to use the security configuration:

a. In the Enterprise Explorer, expand the *Services - Clients* folder in the *lab14ClientWeb* project.

b. Right-click on the *{http://hello}HelloWorldService* entry and choose *Manage Policy Set Attachment* to start the wizard.

c. On the first wizard page, examine the defaults, then press Next.

d. On the next wizard page, press the Add button and select:

```
Endpoint:        all endpoints
Policy set:      Username WSSecurity default
General binding: Client sample
```

Press OK, followed by Ignore and Finish.

_4. Republish the server.

_5. In the *lab14ClientWeb* project, find the servlet and right-click on it and choose *Run As - Run on Server*. The client should run and fail -- we expect failure since we have not configured the service to validate the username/password against some sort of registry on the server. In a real production scenario, we would perform the necessary configuration to establish a user registry, such as LDAP and configure WebSphere to use the registry. Such work is beyond the scope of this course.

However, you should be able to look in the TCP/IP monitor (choose the XML display style) and examine the request message and note the Security header in the message.

Copyright © Descriptor Systems 2014. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.