An Oracle White Paper
June 2011

CA SiteMinder-Oracle Enterprise Gateway Integration Guide

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

1. Overview

This document describes how to configure the Oracle Enterprise Gateway to authenticate and perform authorization via CA SiteMinder R12 Web Access Manager. CA SiteMinder is a centralized Web access management system that provides user authentication and single sign-on, policy-based authorization, identity federation, and auditing of access to Web applications and portals.
This will be demonstrated by the following:

- The OEG Gateway will be configured to authenticate a client via CA SiteMinder via user name and password from either HTTP Basic or a WS-Security username token.
- Upon successful authentication the Gateway will authorize the user for a particular resource via CA SiteMinder.
- The Gateway will be also be configured to add a SiteMinder session token into the HTTP headers or body of the message for authorization without having to authenticate repeatedly via CA SiteMinder.
- A policy will be created that will look for a SiteMinder session token and make a decision depending on whether a SiteMinder session token is present or not. When no token is found in the message or HTTP headers, the user will be authenticated using Basic HTTP via CA SiteMinder. If the session token is found in either the message header or body, the user need not be authenticated again and upon successful validation, the user will be granted access to the web resource.

Sections in this guide:

- Section 1 is an introduction and overview of what this document contains.
- Section 2 explains the setup and configuration of CA SiteMinder.
- Section 3 explains the setup of the resource protected by SiteMinder which is a sample service that ships with the Gateway.
- Section 4 explains the agent configuration required for by the Gateway to successfully connect to CA SiteMinder and the basic policy for authentication and authorization. This section also contains a test of the policy once configured with OEG Service Explorer.
- Section 5 explains the configuration of a policy that will check for an existing SiteMinder session token in the message header. This section also contains a test of the policy once configured with OEG Service Explorer.
- Section 6 explains the configuration of a policy that will check for an existing SiteMinder session token in the message body. This section also contains a test of the policy once configured with OEG Service Explorer.

Setup used for this Guide:

- OEG Gateway 11.1.1.x

- CA SiteMinder Web Access Manager R12 connected to an iPlanet LDAP directory

2. SiteMinder Configuration

Starting the SiteMinder Server

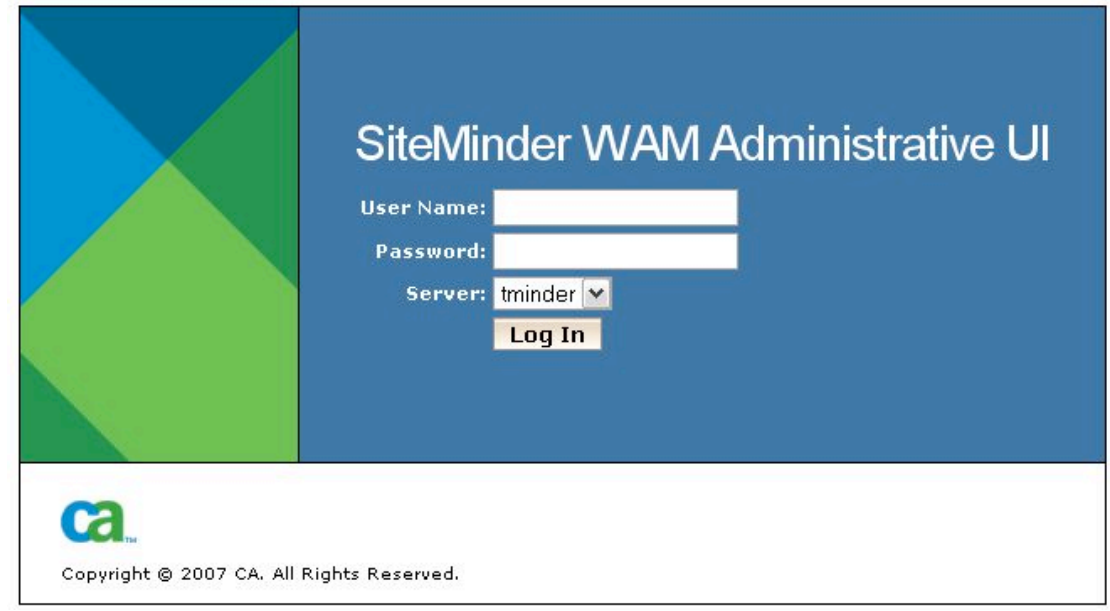
The basics of setting up a SiteMinder Policy will be explained below. However it is strongly recommended that the SiteMinder user guide be consulted.
A copy of SiteMinder can be obtained from CA (www.ca.com).
Starting CA Policy Server on Windows:

- Open **"CA"** program group.
- Select **"IAM Suite"** and then select **"siteminderWAM".**
- Click on **"Start Task Engine".**
- Again from the **"CA"** program group, click on **"SiteMinder Policy Server Management Console".**

Starting SiteMinder Administrator

The SiteMinder WAM Administrator User Interface is used to configure SiteMinder. Log into the Administrator console by using the administrator account credentials.
To log into the SiteMinder WAM Administrative User Interface:

- Open **"CA"** program group**.**
- Select **"IAM Suite"** and select **"siteminderWAM".**
- Click on **"SiteMinder Administrative User Interface".**



Configuring SiteMinder

Step 1: Creating the Host Configuration Object:

- In the SiteMinder Administrative User Interface, click on the **"Infrastructure"** tab.
- Select **"Hosts"** and click on **"Host configuration".**
- Then click on **"Create Host Configuration".**
- In the next screen select the **"Create a new object of type Host Configuration"** radio button option.
- Supply a **"Name"** and **"Description"** in the next screen for the Host. For this guide **"V6HostConfObject"** is used for the agent name.
  - Also click on **"Add"** under the Configuration Values section.
- Under the **"Host"** option enter the IP Address of the SiteMinder Policy server.
- The rest of the options can be left as default.
- Click on **"Submit"** to create the agent.

Step 2: Creating an Agent Configuration Object:

- In the SiteMinder Administrative User Interface, click on the **"Infrastructure"** tab.
- Select **"Agents"** and click on **"Agent configuration"**. For this guide **"V6AgentConfObject"** is used for the agent name.
- Then click on **"Create Agent Configuration".**

- In the next screen select the **"Create a new object of type Agent Configuration"** radio button option.
- Supply a **"Name"** and **"Description"** in the next screen for the agent.
- Also click on **"Add"** under the Parameters section.
- For the **"Name"** field type in **"AgentName"**.
- For the **"Value"** field type in the name of the agent. In this case **"gatewayagent"** is used for this guide.
- Click on **"Submit"** to create the agent.

Step 3: Creating a SiteMinder Realm:

- In the SiteMinder Administrative User Interface, click on the **"Policies"** tab.
- Select **"Domains"** and click on **"Realm".**
- Then click on **"Create Realm".**
- Select a domain from the Domains list before clicking **"Next"** and defining the Realm.
- Supply a **"Name"** and **"Description"** in the next screen for the Realm.
- The Agent that was created previously will now be selected for the agent. Click on the browse button next to the field to browse to the previously created Agent.
- For the **"Value"** field type in the name of the agent.
- The **"Resource Filter"** option specifies the path of the protected resource (web service). For purpose of this guide **/axis/** will be used. This means that all resources after /axis/* is protected. This is the URI of the sample service that ships with the OEG Gateway.
- Default Resource Protection needs to be set to **"Protected".**
- **"Authentication Scheme"** needs to be set to **"Basic".**
- Under the **"Rules"** section click on the **"Create"** tab.
- Supply a **"Name"** and **"Description"** in the next screen for the Rule.
- Under the **"Attributes/Realm and Resource"** section leave the resource as * (will apply to all resources after **/axis/.** (Setting the **"Resource"** to **/axis** implies that SiteMinder will explicitly enforce for that particular resource).
- Under the **"Action"** section select the **"Web Agent actions"** option and select all actions on the Action box.
- Click on **"OK".**
- Also add **"Authentication events"** and **"Authorization events"** leaving the default actions.
- There will now be a total of three rules under the **"Realm".**

NOTE: For step 4 if no domains have been created then a domain needs to be configured before creating the Realm. Settings like the LDAP directories for users, for

example, are specified when creating the domain. Please consult the SiteMinder documentation on how to set up domains in SiteMinder.

Step 4: Creating a Policy for the Domain:

- In the SiteMinder Administrative User Interface, click on the **"Policies"** tab.
- Select **"Domains"** and click on **"Policy".**
- Then click on **"Create Policy".**
- Select the Domain specified in the list the same as the one associated with the Realm created before.
- Supply a **"Name"** and **"Description"** in the next screen for the Policy.
- Click on **"Next".**
- Under the **"Local LDAP Directory"** section click on **"Add Member".**
- When the LDAP connection has been setup correctly under the Domain configuration, there will be a LDAP directory reference that can be selected.

NOTE: For more information on SiteMinder configurations and implementations please refer to the SiteMinder documentation and/or your information technology administrator. The configuration described in the document is a basic SiteMinder Policy configuration and is only used for demonstrative purposes.

3. Setting up the Protected Resource

OEG Gateway ships with an axis sample service. For the purpose of this demonstration it is this service that is protected by SiteMinder.
Details of Sample Service:

- The axis service can be started by running the axissimpleserver.bat file located in the **/Gateway_Install_Dir/win32/bin** for Windows or axissimpleserver.sh located in the **/Gateway_Install_Dir/posix/bin** for Unix/Solaris
- For the purpose of this guide the axis service will be run on the local machine on **port 7070** which is the default port for this service.
- The URL for the resource is: **/axis/services/urn:xmltoday-delayed-quotes**
- The full URL for the service will be: **http://host_ip:7070/axis/services/urn:xmltoday-delayed-quotes**
- The WSDL for the available services can be retrieved from: **http://host_ip:7070/axis/services/urn:xmltoday-delayed-quotes?wsdl**
- The Service home page is**: http://host_ip:7070/**

4. Configuring the OEG Gateway

Configuring the Gateway to interact with CA SiteMinder consists of the following:

- Register the SiteMinder Agent.
- Create the routing policy to the protected resource
- Create the SiteMinder Authentication and Authorization policy.
- Create the relative path for the protected resource connected to the SiteMinder Authentication and Authorization policy.

Register the SiteMinder Agent

In order to act as a PEP (Policy Enforcement Point) for CA SiteMinder, the Gateway must have been set up as a *SOA Agent* with the SiteMinder Policy Server. Therefore the agent needs to be registered on the machine running the Gateway.
Registering the Agent can be done on two ways:

1) Using the **"smreghost"** utility via the command line
2) Via Policy Studio

IMPORTANT: The agent needs to be registered on the machine running the Gateway, so in cases where the Policy Studio is remotely connecting to the Gateway; the **"smreghost"** utility will be used to register the agent configuration via the command line.

Using command line:
For Unix/Solaris

- cd /opt/oeggateway_intall_dir/Linux.i386/lib
- export LD_LIBRARY_PATH=/opt/oeggateway_install_dir/Linux.i386/lib
- Run the smreghost command as in the following example:
- *Example:*

smreghost -i 192.168.0.99 -u SiteMinder -p XXXXXX -hc V6HostConfObject–hn gatewayagent

For Windows:

- cd oeggateway_intall_dir/win32/lib
- Run the smreghost command as in the following example:
- *Example:*

smreghost -i 192.168.0.99 -u SiteMinder -p XXXXXX -hc V6HostConfObject–hn gatewayagent

SMRegHost Usage:

./smreghost -i ipAddress[:port] -u username -p password -hn hostname -hc hostconfigobject

-i <ipAddress[:port]>
-hn <Name for host to be registered>
-hc <Name of host configuration object>

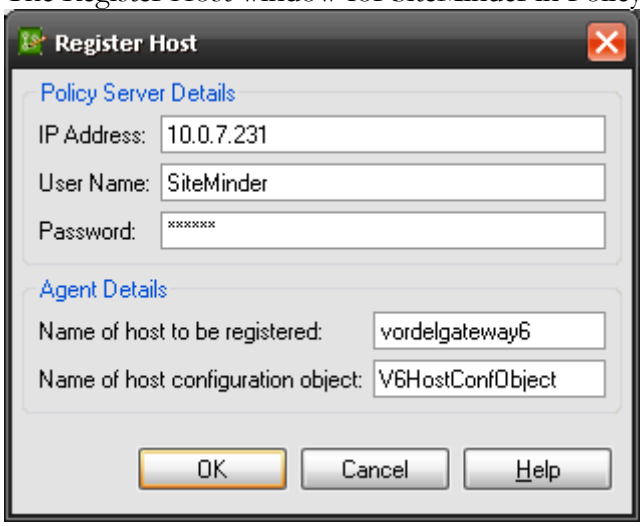
-sh <Shared secret for the host>
-rs (enable shared secret rollover for host)
-u <Administrator username>
-p <Administrator password>
-f <File to store registration data in (defaults to ./SmHost.conf)>
-cp <Name of crypto provider (BSAFE or PKCS11)>
-cd <Path to crypto provider DLL or config file>
-ct <Crypto provider token label>
-ck <Crypto provider token PIN>
-o <Overwrite existing Trusted Host>

- Once the command has run successfully, a file called **"smhost.conf"** will be created in the /opt/oeggateway_install_dir/Linux.i386/bin folder.
- Browse to the VXP Web Administration Interface on https://appliance_IP:10000/.
- Click on the **"Upload and Download"** option and download the **"smhost.conf"** file to the machine running the Policy Studio.
- Finally, use the **"Browse"** when configuring the agent in Policy Studio as described below instead of the **"Register"** option to import the configuration.

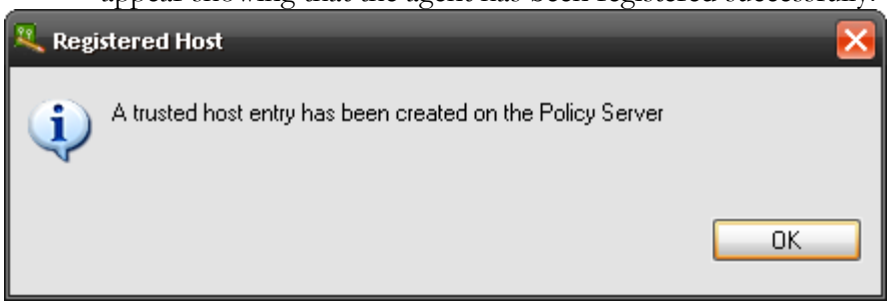
Registering the agent using Policy Studio:

- Start Policy Studio by running **"policystudio.exe"** (Windows) or **"policystudio.sh"** (Unix/Solaris) from the Policy Studio root directory.
- Click on the URL for the Gateway or Policy Director if the Gateway/s is managed via Policy Director.
- Click on the OEG Gateway process listed to open the configuration window in a new tab.
- Click on the **"External Connections"** module.
- Right click on **"SiteMinder/SOA Security Manager Connections"** and select **"Add SiteMinder Connection".**
- Click on the **"Register"** button OR **"Browse"** button if the agent configuration files has already been created via the command line as explained above.
- Enter the **"IP Address"** of the SiteMinder Policy Server.
- Enter the **"User Name"** and **"Password"** of the account used to connect to the SiteMinder Policy Server.
- Under **"Agent Details"** enter a value for the **"Name of the host to be registered".** This can be ANY descriptive value and should not already exist under the **"Trusted Host"** list on the SiteMinder Policy Server. For this guide **"vordelgateway6"** is used.
- Enter the **"Name of the host configuration object"** as is configured in section 2.3 Configuring SiteMinder step1. For this guide it is **"V6HostConfObject".**

The Register Host window for SiteMinder in Policy Studio

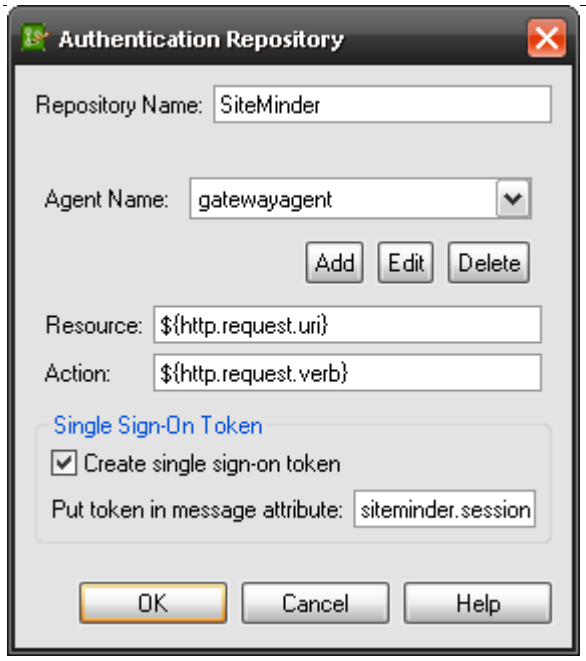


- Once all the details have been provided and click on **"OK"** and a dialogue will appear showing that the agent has been registered successfully.



- Expand the **"Authentication Repository Profiles"** object under **the "External Connections"** tree.
- Right click on **"CA SiteMinder Repositories"** and select **"Add a new Repository".**
- Enter a name for the repository. For this guide **"SiteMinder"** is used.
- Select the configured agent in the drop down menu for the **"Agent Name"** value.
- The rest of the options can be left default. Click on the help button if more details are required for the configuration options here.

The SiteMinder Authentication Repository for screen in Policy Studio:



Create the Routing Policy to the Protected Resource

- In the Policy Studio configuration window for the Gateway process click on the **"Policies"** module.
- Right click on "Policies" at the top node and select "Add Container".
- Name the container "Routing".
- Right click on **"Routing"** container and select **"Add Policy".**
- Name the policy to **"Route to Axis Service".**
- Drag a **"Connect to URL" filter** from the **"Routing"** category.
- Rename the Name of the filter to: **Route to Axis Service**
- Enter URL: **http://host_ip:7070/axis/services/urn:xmltoday-delayed-quotes**
- Click on **"Finish".**
- This policy will be called to via a **"Policy Shortcut"** filter that will be configured in the next section as part of the SiteMinder Authentication and Authorization policy.

Create the SiteMinder Authentication and Authorization Policy

The policy is going to authenticate and authorize a user that exists in an LDAP directory server via SiteMinder.

The flow of the policy:

- User will present a username and password combination to the Gateway using HTTP Basic authentication. The user's credentials will be passed to CA SiteMinder, which will authenticate the user.
- The user will be authorized via CA SiteMinder for a particular resource
- The request will be routed to the resource (axis web service) via the policy shortcut that will call the **"Route to Axis Service"** routing policy.

Step 1: Configure a HTTP Basic Authentication Filter

- In the Policy Studio configuration window for the Gateway process click on the **"Policies"** module.
- Right click on **"Policies"** and select **"Add Policy".** Name the policy **"SiteMinder".**
- Click on the policy and add a **"HTTP Basic"** filter located in the **"Authentication"** filter category located on the right hand side of Policy Studio.
- **HTTP Basic Filter Configuration:**
- **Name:** of the filter can be left default or changed to any descriptive name.
- **Realm:** Populated automatically by the value specified in System Settings in the Gateway.
- **Credential Format:** select User Name from the drop down list.
- **Repository Name:** Select **"SiteMinder"** repository from the drop down field.
- Click on **"OK"**

Step 2: Configure a SiteMinder Authorization Filter

- Drag an **"Authorization" filter** from the **"CA SiteMinder"** filter category.
- Name the filter **"Authorize User via SiteMinder".**
- Settings can be left as default.
- Click on **"Finish"**.
- Connect the **"HTTP Basic"** filter configured in step 1 to the **"CA Authorization"** filter to the with a success path.

Step 3: Configure a Policy Shortcut to call point to Route to Web

- Drag a **"Policy Shortcut'** filter from the **"Utility"** filter category.
- Choose the **"Route to Axis Service"** policy that will route the message to the Axis web service.
- Click on **"Finish"**.
- Connect the **"CA Authorization"** to the **"Policy Shortcut"** filter with a success path.

Step 4: Create a New Relative Path

- Click on the **"Services"** module in Policy Studio.
- Expand **"Processes"**, **"OEG Gateway"** and right click on the **"Default Services".**
- Select "Add Relative Path" and enter: **/axis/services**
- Deploy the new configuration to the server by pressing **"F6"** on the keyboard or by clicking on **"Settings"** at the top and select **"Deploy"**.

Step 4: Test Policy with OEG Service Explorer

OEG Service Explorer is a free stress and security tool for web services developed by Vordel. It will be used as the client for testing.
OEG Service Explorer can generate a request via the available web service:

- Start **OEG Service Explorer** by running **"OEG Service Explorer.exe"** (win32) or **"OEG Service Explorer.sh"** (UNIX) located in the OEG Service Explorer root directory.
- Click on the **"Import WSDL"** button on the top tool bar.
- Select the **"WSDL URL"** option and enter the URL of the Axis Service WSDL.
- If running the Axis Service locally as described in section 3 the URL is: http://localhost:7070/axis/services/urn:xmltoday-delayed-quotes?wsdl
- Click on **"OK".**
- A request will be automatically generated.

Alternatively the following request can be used:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body>
        <ns:getQuote

soap:encodingStyle="http://schemas.xmlsoap.org/soap/encodi
ng/"
            xmlns:ns="http://stock.samples" />
    </soap:Body>
</soap:Envelope>
```

- Configure the URL that the request needs to be sent to by clicking on the configuration block underneath the top menu.
- Enter the URL for the XML Gateway and resource path. In this case it is:
- **http://gateway_host_ip:8080/axis/services**
- Copy the test message above into the Soap Request window if it has not been autogenerated already.
- Click on **"Security"** tab followed by the **"HTTP Authentication"** tab.
- Choose **'HTTP Basic'** and enter the username and password of the user will be authenticated via SiteMinder. For purpose of this guide it is a user called Hubert Farnsworth that is located in the LDAP directory that SiteMinder is connected to.
  - Username: cn=hubert farnsworth,o=planet express,l=new york,st=ny,c=us
  - Password: goodNews

  NOTE: The username in this case is actually the user's distinguished name (DN) in the LDAP directory.
- Click on **"Run"** to send the message.
- The request will be sent via the Gateway that in turn will contact CA SiteMinder to authenticate and authorize the user.
- Once successfully completed the Gateway will send the request on to the protected resource.

5. Retrieving a SiteMinder Session Token from the Message Header

In the previous section a policy has been configured to authenticate and authorize the request.

In this section a policy will be created that will look for an existing SiteMinder session token in the message headers. When a SiteMinder session token is found in the message (header or body); the user does not have to be authenticated again.

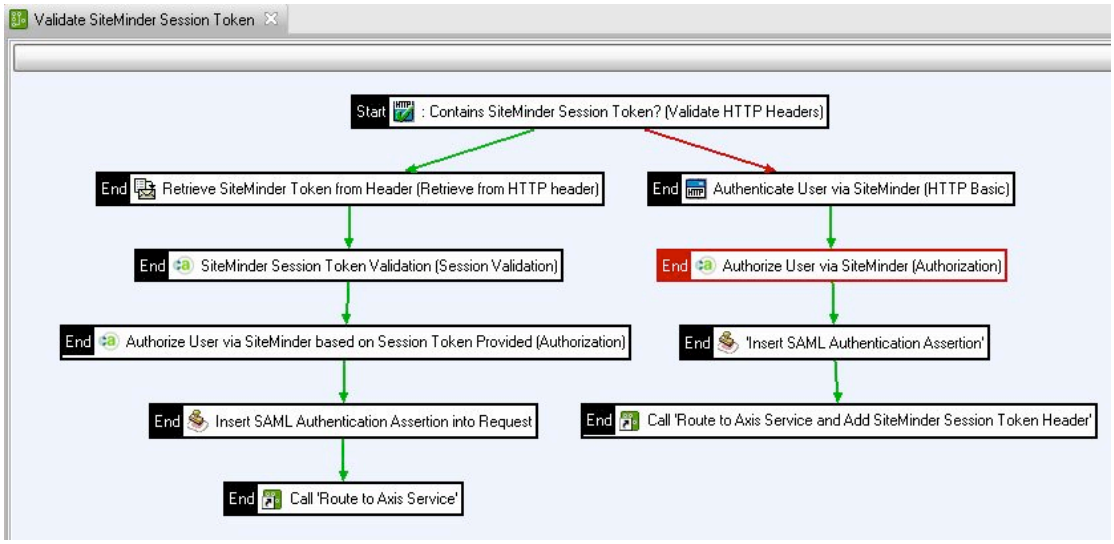
In a real world deployment scenario, it is not practical to have to authenticate and authorize each user for every request they send. By inserting a SiteMinder session token into a message after successfully authenticating a user the token in the message is validated instead of re-authenticating the user for every following request. A SAML Authorization Assertion will also be inserted for consumption by a downstream web service.

The flow of the policy will be as follows:

- Check the request's HTTP headers for the presence of a SiteMinder session token.

- When the token is present, the "Retrieve from HTTP Header" filter is run to retrieve the value of the token from the "siteminder.session" attribute.
- When the attribute containing the token has been retrieved successfully the session will be validated using a Ca SiteMinder "Session Validation" filter to check if the user token is still valid.
- When the token has been successfully validated then the user will be authorized using the CA SiteMinder "Authorization" filter.
- A SAML Authentication Assertion will be inserted into the message for consumption by downstream web services.
- The message will then be routed to the Axis service.
- If the Message did not have the "siteminder.session" token in the header and passed through the "HTTP Basic" filter the user will be authorized based on user credentials.
- The user will be authorized using the CA SiteMinder "Authorization" filter.
- The message will pass through a "SAML Authentication Assertion" filter and inserted into the message for consumption by downstream web services.
- The message will then be passed to the "Route to Axis Service" policy shortcut that will route the message to the protected resource. The response message will pass through an "Add HTTP Header" filter that will add a HTTP Header to the message response containing a SiteMinder session token.

The policy once configured:



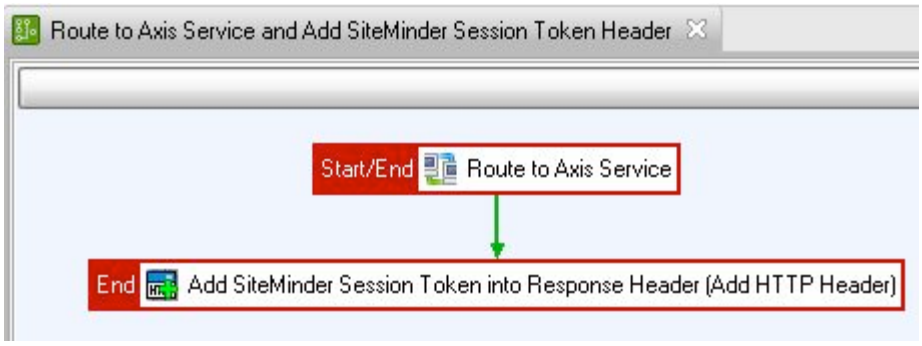
Configure a Routing Policy that will Add a SiteMinder Token to the HTTP Headers of the Response

The first step will be to create a new routing policy that will add the SiteMinder session token into the HTTP headers of the response request.

Creating the "Route to Axis Service and Add SiteMinder Token to Header" Policy:

- Navigate to the **"Route to Axis Service"** policy as configured in section 4.2.
- Right click on the **"Route to Axis Service**" on the left and click on **"Copy".**
- Right click on the **"Routing"** container and select **"Paste"**
- Rename the policy to: **Route to Web Service and Add SiteMinder Token to Header.**
- Click on **"OK".**
- Add an **"Add HTTP Header"** filter from the **"Conversion"** category.
- Name the filter **"Add SiteMinder Session Token into Response Header".**
- For the **"HTTP Header Name"** type **"SiteMinderToken"**.
- For the **"HTTP Header Value"** type **${siteminder.session}**. This is the attribute value that is specified by the SiteMinder Authentication repository configuration as configured in section 4.1. When a user is successfully authenticated to SiteMinder, this message attribute will contain the user's SiteMinder session token.
- Click on **"Finish"**.
- Connect the **"Route to Axis Service"** filter to the **"Add SiteMinder Session Token into Response Header"** filter via a success path.

The routing policy will now look like below:



Configuring the Policy that will Validate SiteMinder Session Tokens located in the HTTP Headers of a Request

Step 1: Configure a "Validate HTTP Headers" filter to check if a SiteMinder Session Token is present in the HTTP Headers of the Message.

- Click on the **"Policies"** module in the tree on the left hand side of Policy Studio.
- Right click and select **"Add Policy".**
- Name the Policy **"Validate SiteMinder Session Token in HTTP Headers".**
- Drag a **"Validate HTTP Headers"** filter from the **"Content Filtering"** category.
- Name the filter **"Contains SiteMinder Session Token?"**
- Under the **"Enter a Regular Expression"** section click on the **"Add"** button.
- A Window titled **"Configure Regular Expression"** will open. Enter **"SiteMinderToken"** in the name field and select the **"Required"** radio button.
- Click on **"Ok".**
- Right click on this filter and select **"Set as Start".**

Step 2: Configure a "Retrieve from Attribute" filter that will retrieve the existing token from the HTTP Header of the message.

- The **"Retrieve from HTTP Header"** filter is located in the **"Attributes"** group.
- Name the filter **"Retrieve SiteMinder Token from Header".**
- In the **'HTTP Header Name'** field enter **"SiteMinderToken"**. In the **'Attribute ID'** field enter **"siteminder.session".**
- Click on **"Finish".**
- Connect the **'Validate HTTP Header'** filter to the **"Retrieve from HTTP Header"** filter with a success path.

Step 3: Configure a SiteMinder "Session Validation" filter for validation of the SiteMinder Session Token.

- Add a **"Session Validation"** filter located in the **"CA SiteMinder"** category.
- For the **"Agent Name"** select the SiteMinder agent that was created in section 4.1 from the drop down field. For this guide it is **"gatewayagent"**.
- Ensure that the **"Message Attribute Containing Session"** is set to **"siteminder.session"**.
- The rest can be left default.
- Click on **"Finish".**
- Connect the **"Retrieve from HTTP Header"** filter to the **"Session Validation"** filter with a success path.

Step 4: Configure a "Authorization" filter that will Authorize the Client for the Protected Resource.

- Drag an **"Authorization" filter** from the **"CA SiteMinder"** category**.**
- Name the filter **"Authorize User via SiteMinder based on Session Token".**
- Settings can be left as default.
- Click on **"Finish"**.
- Now connect the **"Session Validation"** filter to the **"Authorization"** filter with a success path.

Step 5: Configure an "Insert SAML Authentication Assertion" filter that will add a SAML Authentication Assertion to the Message for Consumption by Downstream Service. (This is done for demonstrative purposes only and is not necessary for purpose of this guide)

- The **"Insert SAML Authentication Assertion" filter** is located in the **"Authorization"** category.
- **Expiry Date:** Set to any desired value.
- **SOAP Actor/Role:** Choose **"Current Actor/Role Only"** from the drop down list.
- On the **"Sign Assertion"** Tab select **"No Signature with Assertion"** and select any value from the drop down field for **"Issuer Name".**
- Under "**Advanced Options"** tick "**Insert SAML Attribute Statement"** and **"Indent".**
- The rest of the options could be left default.
- Click on **"Finish".**

Step 6: Configure a "Policy Shortcut" filter and link it to the "Route to Axis Service" policy.

- Drag a **"Policy Shortcut"** filter from the **"Utility"** category.
- Choose the **"Route to Web Service and Add SiteMinder Token to Header"** policy that will route the message to the sample web service and add the session token to the header of the response.
- Click on **"Finish".**
- Connect the **"Policy Shortcut"** filter to the **"Insert SAML Authentication Assertion" filter** with a success path.

Configuring the failure path filters:

Starting back at the top of the policy the failure path will now be configured.

Step 1: Create a HTTP Basic Authentication filter to Authenticate the User via SiteMinder

- The **"HTTP Basic"** filter used in the first **"SiteMinder"** policy can be used for this policy so simply copy and paste the **"HTTP Basic"** filter into the policy canvas.
- Rename the filter to **"Authenticate User via SiteMinder".**
- Connect the **"Validate HTTP Headers"** filter to the **"HTTP Basic"** filter with a failure path.

Step 2: Configure a "SiteMinder Authorization" filter that will Authorize the Client

- Add an **"Authorization" filter** so simply make a copy of the **"Authorization"** filter configured already in the previous **"SiteMinder"** policy and copy it into the policy canvas
- Rename the filter to **"Authorize User via SiteMinder".**
- Connect the **"HTTP Basic"** filter to the **'Authorization'** filter with a success path.

Step 3: Configure an "Insert SAML Authentication Assertion' filter that will add a SAML Authentication Assertion to the message for consumption by downstream service.

- Copy the **"Insert SAML Authentication Assertion"** filter created above and connect it to the **"Authorization"** filter with a success path

Step 4: Configure a "Policy Shortcut" filter and link it to the "Route to Axis Service and Add SiteMinder Token" policy.

- Drag a **"Policy Shortcut" filter** from the **"Utility"** group in the filter palette.
- Choose the **"Route to Web Service and Add Token into Header of Response"** policy that will route the message to the sample web service and also add the SiteMinder session token into the HTTP header of the message.
- Click on **"Finish".**

Step 5: Create a New Relative Path

- Click on the **"Services"** module in Policy Studio.
- Expand **"Processes", "OEG Gateway"** and right click on the **"Default Services".**
- Select **"Add Relative Path"** and enter: **/axis/validation**
- Select the **"Validate SiteMinder Session Token in HTTP Headers"** policy.
- Click on **"OK".**
- Deploy the new configuration to the server by pressing **"F6"** on the keyboard or by clicking on **"Settings"** at the top and select **"Deploy"**.

Step 6: Testing the Retrieve from Header Policy with OEG Service Explorer

To test the modified policy the same procedure will be used as before. However, a message will also be sent through that already contains the SiteMinder session token in the header of the message to demonstrate how the policy decision tree works.

OEG Service Explorer can generate a request via the available web service:

- Start **OEG Service Explorer** by running OEG Service Explorer.exe (win32) or OEG Service Explorer.sh (UNIX) located in the OEG Service Explorer root directory.
- Click on the **"Import WSDL"** button on the top tool bar.
- Select the **"WSDL URL"** option and enter the URL of the Axis Service WSDL.
- If running the Axis Service locally as described in section 3 the URL is: http://localhost:7070/axis/services/urn:xmltoday-delayed-quotes?wsdl
- Click on **"OK".**
- A request will be generated formatted to what the service expects.

Alternatively the following request can be used:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body>
        <ns:getQuote

soap:encodingStyle="http://schemas.xmlsoap.org/soap/encodi
ng/"
            xmlns:ns="http://stock.samples" />
    </soap:Body>
</soap:Envelope>
```

- Configure the URL that the request needs to be sent to by clicking on the configuration block underneath the top menu.
- Enter the URL for the XML Gateway and resource path. In this case it is:
- **http://gateway_host_ip:8080/axis/validation**
- Copy the test message above into the Soap Request window if it has not been auto generated already.
- Click on **"Security"** tab followed by the **"HTTP Authentication"** tab.

- Choose **'HTTP Basic'** and enter the username and password of the user will be authenticated via SiteMinder. For purpose of this guide it is a user called Hubert Farnsworth that is located in the LDAP directory that SiteMinder is connected to.
  - Username: cn=hubert farnsworth,o=planet express,l=new york,st=ny,c=us
  - Password: goodNews

  NOTE: The username in this case is actually the user's distinguished name (DN) in the LDAP directory.
- Click on **"Run"** to send the message.
- The request will follow the failure path of the **"Validate SiteMinder Session Token"** policy.
- The response will contain a SiteMinder Session Token in the HTTP headers.

The next message that will be sent through will be sent with a SiteMinder session token inserted into the header.

- Copy the value from the **"siteminder.session"** attribute as printed out in the trace console. Please see **"Appendix"** for more details on how to do this.
- Enter the URL for the Gateway and resource path. In this case: **http://GATEWAY_HOST:8080/axis/validation**
- Copy the test message into the SOAP Request window if it has not been auto generated.
- Click on the **"Headers"** tab and click on **"Add"**.
- In the **"Name"** field enter **"SiteMinderToken"** and in the **"Value"** field paste the string that was copied above. This is actually the same session value that was inserted and viewed in the **"Headers"** tab of the returned message.
  - Click on **"Security"** tab followed by the **"HTTP Authentication"** tab.
  - Choose **'HTTP Basic'** and enter the username and password of the user will be authenticated via SiteMinder. For purpose of this guide it is a user called Hubert Farnsworth that is located in the LDAP directory that SiteMinder is connected to.
    - Username: cn=hubert farnsworth,o=planet express,l=new york,st=ny,c=us
    - Password: goodNews

    NOTE: The username in this case is actually the user's distinguished name (DN) in the LDAP directory.
  - Click on **"Run"** to send the message.
  - The request will follow the success path of the **"Validate SiteMinder Session Token"** policy as the user has been validated successfully via the token in the HTTP Headers of the request.

Section 5 Summary:

In this section the policy demonstrates how the message can follow two different paths depending on whether a SiteMinder token is present in the header or not.
Success Path: Token is present in the http header of message

- Message will be checked for the presence of a valid token located in the HTTP header of the request.
- Token will be retrieved from header of message.
- The SiteMinder session token will be validated.
- The user will be authorized.
- The message will pass through an "Insert SAML Authentication Assertion" filter for consumption by a downstream web service.
- The message will then be routed to the sample web service.

Failure Path: Token is not present in the http header of message

- Message will be checked for the presence of a valid token.
- If the token is not present then the message will follow the failure path to the "HTTP Basic" filter where the user will be authenticated if a valid username and password are present.
- When the user credentials have been successfully authenticated the message will pass to the CA SiteMinder "Authorization" filter that will authorize the user.
- The "siteminder.session" attribute will be copied from the message and modified to be used by the "Insert SAML Authentication Assertion" filter and place it in the message as part of the SAML insertion for consumption by a downstream web service.
- The message will then be routed to the modified 'Route to Web Service and Add Token into Header of Response' policy shortcut which will route the message to the sample web service and also add the session token into the HTTP Header of the response message.

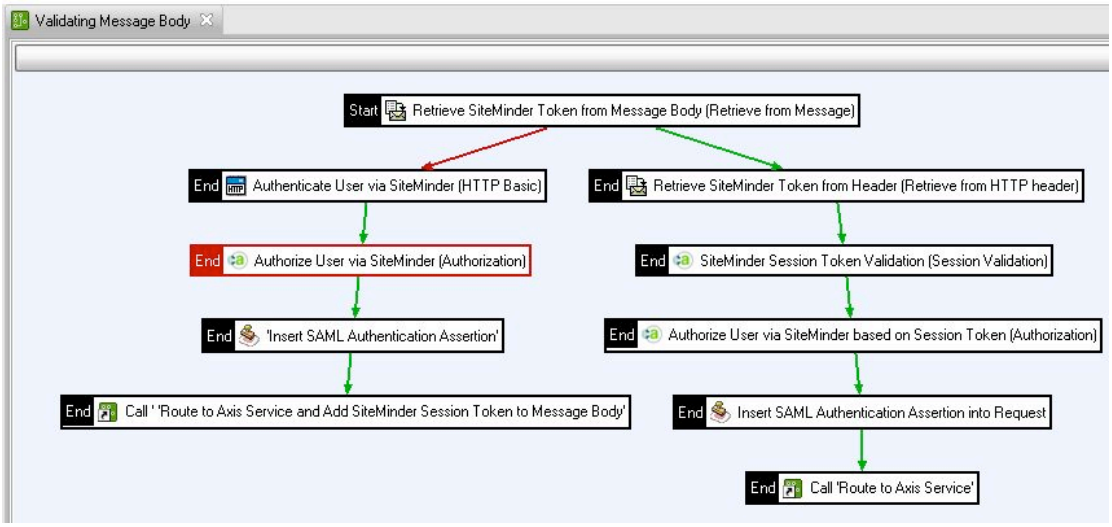
6. Retrieving a SiteMinder Session Token from the Message Header

Similar to the policy above it is also possible to add to or retrieve the SiteMinder session token from the message body instead of the HTTP header's.

The flow of the policy:

- Retrieve the session token from the SOAP Header in the body of the message when present
- If the session token attribute has been retrieved the attribute will be validated via the "Validate Attribute" filter.
- Once validated it will pass it to the CA SiteMinder "Session Validation" filter to validate the session token in the message body.
- When the token has been successfully validated, the user will be authorized using the CA SiteMinder "Authorization" filter.
- The message will then be passed through an "Insert SAML Authentication Assertion" filter where a SAML Authentication Assertion will be inserted into the message for consumption via a downstream web service.
- The message will then be routed to the protected resource.
- If the message body (SOAP Header) did not contain a session token the message will be passed to the 'HTTP Basic' filter where the user will be authenticated using valid user credentials (username and password).
- The message will then pass through the CA SiteMinder "Authorization" filter where the user will be authorized to access the protected resource.
- The message will then be passed through an "Insert SAML Authentication Assertion" filter where a SAML Authentication Assertion will be inserted into the message for consumption via a downstream web service.
- The message will then be passed to the Route to Web Service policy shortcut that will route the message to the web service URL. The response will then pass through an XML enrichment filter (Insert XML Node filter) that will add a SOAP Header node into the message body of the response message containing the SiteMinder session token.

The Policy as described above:



Configuring a Routing Policy that will Add a SiteMinder Token into the Body of the Response

Creating the Routing Policy:

- Navigate to the **"Route to Axis Service"** policy as configured in section 4.2.
- Right click on the **"Route to Axis Service"** on the left and click on **"Copy".**
- Right click on the **"Routing"** container and select **"Paste".**
- Rename the policy to: **Route to Web Service and Add SiteMinder Token to Message Body.**
- Click on **"OK".**
- Drag an **"Insert XML Node" filter** from the **"Conversion"** filter category.
- For the "**Name"** enter: **Add Token into SOAP Header of Message**
- Under the **"Configure Where to Insert the New Node"** section for the **"XPath Location"** click on **"Add"**.
  - For the Name enter: **Add SOAP Header with SiteMinderToken**
  - For **"XPath Expression"** enter: **/soap:Envelope**
  - For **"Prefix"** enter: **soap**
  - For **"URI"** enter: **http://schemas.xmlsoap.org/soap/envelope/**
- Click on **"OK"**.
- Back on the main filter configuration screen choose **"Append"** under the **"Configure Where to Insert the New Node".**
- Under **"Configure New Node Details"** for **"Node Type"** select **"Element".**
- For **"Node Content"** enter:

<soap:Headerxmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
- <SiteMinderToken>${siteminder.session}</SiteMinderToken></soap:Header>
- Click on **"Finish".**

Configuring the Policy that will Validate SiteMinder Session Tokens located in the Message Body of the Request

Step 1: Configure a "Retrieve Attributes from Message" filter

- Right Click on "**Policies"** module in the tree on the left hand side of Policy Studio.
- Right click on policies and select **"Add Policy".**
- Name the Policy **"Retrieve SiteMinder Token from Message Body".**
- Drag a **"Retrieve Attributes from Message"** filter from the **"Attributes"** category.
- For **"Attribute Location"** click on **"Add".**
- Enter **"SiteMinderToken"** for the name.
- For the **"XPath Expression"** click on the button next to the **"XPath Expression"** field.
- The **"XPath Expression Wizard"** can be used to auto-generate an XPath expression based on a node selected in a message.
- The <SiteMinderToken> node has been selected, which contains the SiteMinder session token.
- The XPath entry is: /soap-env:Envelope/soap:Header/SiteMinderToken
- With the path displayed in the **"Selected"** field at the bottom, click on the **"Use this Path"** button. Then click on **"OK".**
- The auto-generated XPath expression will now be configured as the XPath expression back on the **"Enter XPath Expression"** window as displayed below:
- The XPath expression is: /soap-env:Envelope/soap:Header/SiteMinderToken
- Click on **"OK".**
- This will bring up the **"Retrieve from Message"** main window. Select **"Extract the Content of the Node".**
- **"Attribute ID"** needs to be set to **"siteminder.session".** (without quotes)
- Click on **"Finish".**
- Right click on the filter and select **"Set as Start".**

Step 2: Configure a "Validate Message Attribute" filter

- Add a **"Validate Message Attribute" filter** located in the **"Content Filtering"** category.
- Under **"Enter Regular Expression"** click on **"Add".**
- For **"Name"** enter "**siteminder.session"** and choose the **"Required"** option.

- Click on **"OK"**.
- Click on **"Finish"**.

Step 3: Configure a SiteMinder "Session Validation" filter for validation of the SiteMinder Session Token.

- Add a **"Session Validation"** filter located in the **"CA SiteMinder"** category.
- For the **"Agent Name"** select the SiteMinder agent that was created in section 4.1 from the drop down field. For this guide it is **"gatewayagent"**.
- Ensure that the **"Message Attribute Containing Session"** is set to **"siteminder.session"**.
- The rest can be left default.
- Click on **"Finish".**
- Connect the **"Retrieve from HTTP Header"** filter to the **"Session Validation"** filter with a success path.

Step 4: Configure an "Authorization" filter that will Authorize the Client for the Protected Resource.

- Drag an **"Authorization" filter** from the **"CA SiteMinder"** category**.**
- Name the filter **"Authorize User via SiteMinder based on Session Token".**
- Settings can be left as default.
- Click on **"Finish"**.
- Now connect the **"Session Validation" filter** to the **"Authorization"** filter with a success path.

Step 5: Configure an "Insert SAML Authentication Assertion" filter that will add a SAML Authentication Assertion to the Message for Consumption by Downstream Service. (This is done for demonstrative purposes only and is not necessary for purpose of this guide)

- The **"Insert SAML Authentication Assertion" filter** is located in the **"Authorization" group**.
- Configure the filter as follows:
- **Expiry Date:** Set to any desired value.
- **SOAP Actor/Role:** Choose **"Current Actor/Role Only"** from the drop down list.
- On the **"Sign Assertion"** Tab select **"No Signature with Assertion"** and select any value from the drop down field for **"Issuer Name".**
- Under "**Advanced Options"** tick **"Insert SAML Attribute Statement"** and **"Indent".**
- The rest of the options could be left default.

- Click on **"Finish".**

Step 6: Configure a "Policy Shortcut" filter and link it to the "Route to Axis Service" policy.

- Drag a **"Policy Shortcut"** filter from the **"Utility"** category.
- Choose the **"Route to Web Service and Add SiteMinder Token to Header"** policy that will route the message to the sample web service and add the session token to the header of the response.
- Click on **"Finish".**
- Connect the **"Policy Shortcut"** filter to the **"Insert SAML Authentication Assertion" filter** with a success path.
- Refresh the server by pressing **'F6'** on the keyboard or by clicking on "Settings" at the top and select **"Deploy"**.

Configuring the failure path filters:

Starting back at the top of the policy the failure path will now be configured.

Step 1: Create a HTTP Basic Authentication filter to Authenticate the User via SiteMinder

- The **"HTTP Basic"** filter used in the first **"SiteMinder"** policy can be used for this policy so simply copy and paste the **"HTTP Basic"** filter into the policy canvas.
- Rename the filter to **"Authenticate User via SiteMinder".**
- Connect the **"Validate HTTP Headers"** filter to the **"HTTP Basic"** filter with a failure path.

Step 2: Configure a "SiteMinder Authorization" filter that will Authorize the Client for the Protected Resource.

- Add an **"Authorization" filter** so simply make a copy of the **"Authorization"** filter configured already in the previous "SiteMinder" policy and copy it into the policy canvas
- Rename the filter to **"Authorize User via SiteMinder".**
- connect the **"HTTP Basic"** filter to the **'Authorization'** filter with a success path.

Step 3: Configure an "Insert SAML Authentication Assertion' filter that will add a SAML Authentication Assertion to the message for consumption by downstream service.

- Copy the **"Insert SAML Authentication Assertion"** filter created above and connect it to the **"Authorization"** filter with a success path. Refer to policy diagram above if necessary.

Step 4: Configure a "Policy Shortcut" filter and link it to the "Route to Axis Service and Add SiteMinder Token" policy.

- Drag a **"Policy Shortcut" filter** from the **"Utility"** group in the filter palette.
- Choose the **"Route to Web Service and Add Token into Header of Response"** policy that will route the message to the sample web service and also add the siteminder session token into the HTTP header of the message.
- Click on **"Finish".**
- When these filters have all been connected as described it will look like the policy diagram above.
- Refresh the server by pressing **"F6"** on the keyboard or by clicking on settings at the top and select **"Deploy"**.

Step 5: Create a New Relative Path

- Click on the **"Services"** module in Policy Studio.
- Expand **"Processes", "OEG Gateway"** and right click on the **"Default Services".**
- Click on the **"/axis/validation"** relative path
- Change the policy it is routing to the **"Validate SiteMinder Session Token in Message Body".**
- Deploy the new configuration to the server by pressing **"F6"** on the keyboard or by clicking on **"Settings"** at the top and select **"Deploy"**.

Step 6: Testing the Retrieve from Message Policy with OEG Service Explorer

To test the modified policy the same procedure will be used as before. However, a message will also be sent through that already contains the SiteMinder session token in the body of the message to demonstrate how the policy decision tree works.

OEG Service Explorer can generate a request via the available web service:

- Start **OEG Service Explorer** by running **"OEG Service Explorer.exe"** (win32) or **"OEG Service Explorer.sh"** (UNIX) located in the OEG Service Explorer root directory.
- Click on the **"Import WSDL"** button on the top tool bar.
- Select the **"WSDL URL"** option and enter the URL of the Axis Service WSDL.
- If running the Axis Service locally as described in section 3 the URL is: http://localhost:7070/axis/services/urn:xmltoday-delayed-quotes?wsdl
- Click on **"OK".**
- A request will be generated formatted to what the service expects.

Alternatively the following request can be used:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body>
        <ns:getQuote

soap:encodingStyle="http://schemas.xmlsoap.org/soap/encodi
ng/"
            xmlns:ns="http://stock.samples" />
    </soap:Body>
</soap:Envelope>
```

- Configure the URL that the request needs to be sent to by clicking on the configuration block underneath the top menu.
- Enter the URL for the XML Gateway and resource path. In this case it is:
- **http://gateway_host_ip:8080/axis/validation**
- Copy the test message above into the Soap Request window if it has not been auto generated already.
- Click on **"Security"** tab followed by the **"HTTP Authentication"** tab.
- Choose **'HTTP Basic'** and enter the username and password of the user will be authenticated via SiteMinder. For purpose of this guide it is a user called Hubert Farnsworth that is located in the LDAP directory that SiteMinder is connected to.
    - Username: cn=hubert farnsworth,o=planet express,l=new york,st=ny,c=us
    - Password: goodNews

  NOTE: The username in this case is actually the user's distinguished name (DN) in the LDAP directory.
- Click on **"Run"** to send the message.
- The request will follow the failure path of the **"Validate SiteMinder Session Token"** policy.
- The response will contain a SiteMinder Session Token in the message body.

The next message to be sent through will be a message modified to include the session token in the SOAP Header of the message body.

The test message:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Header>
        <SiteMinderToken>
        INSERT_SITEMINDER_SESSION_TOKEN_HERE
        </SiteMinderToken>
</soap:Header>
<soap:Body>
<ns:getQuote
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encodi
ng/" xmlns:ns="http://stock.samples"/>
</soap:Body>
</soap:Envelope>
```

Insert the SiteMinder session token that was generated in the response of the first test message sent through using the user credentials (username and password). Insert the token between the <SiteMinderToken> elements of the message and send the request again.
This time the request will follow the success path of the policy as it contains the SiteMinder session token in the message body.

Section 6 Summary:
In this section the policy demonstrates how the message can follow two different paths depending on whether a SiteMinder token is present in the message or not.
Success Path: Token is present in the message body:

- Token will be retrieved from body of message.
- The attribute will be validated.
- The SiteMinder token will be validated.
- The user will be authorized.
- The message will then be passed through an "Insert SAML Authentication Assertion" filter where a SAML Authentication Assertion will be inserted into the message for consumption via a downstream web service.
- The message will then be forwarded to the protected web service.

Failure Path: Token is not present in the message body:

- If the token is not present then the message will follow the failure path to the HTTP Basic filter where the user will be authenticated when a valid username and password are present.
- When the user has been successfully authenticated, the user will be authorized.
- After successful authorization the message will pass through the "SAML Authentication Assertion" filter which will then insert a SAML Authentication Assertion into the message for consumption by a downstream web service.
- The message will then be routed to the "Route to Axis Service and Add Token into Message Body" policy shortcut which will route the message to the protected resource and also add the session token into the SOAP Header of the message body of the response message.

7. Conclusion

This document demonstrated how to configure the OEG Gateway to authenticate and authorize users via CA SiteMinder. It also demonstrated how the session token can be used to authorize the user without having to authenticate the user again which demonstrates the single sign on capability provided by integrating OEG Gateway with CA SiteMinder.
This configuration can be part of a larger policy, including features such as XML threat detection and conditional routing, features which are out of the scope of this document but are covered in other documents which can be obtained from Oracle at http://www.oracle.com.

8. Appendix

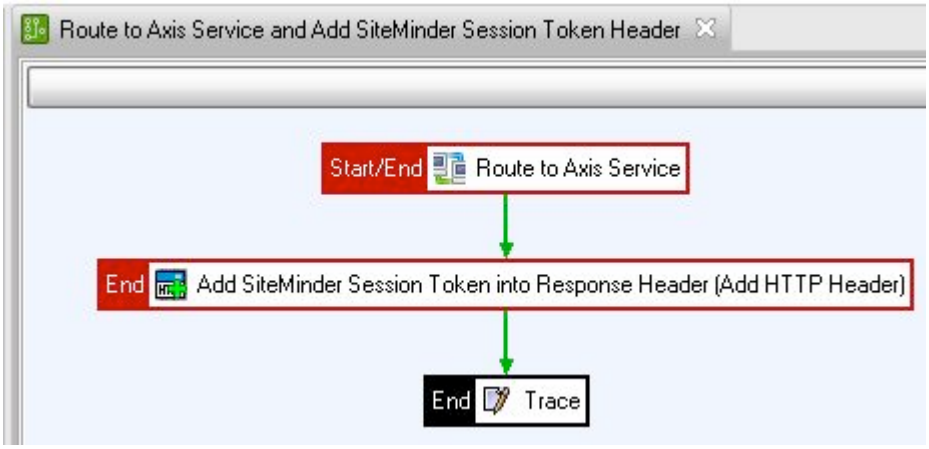
Adding a Trace Filter for displaying Verbose Attribute information in the Trace Console

To be able to get hold of the SiteMinder session token, a "Trace" filter will be added to one of the configured routing policies, which will then output all attributes available in the trace console.
Add a Trace Filter to a Policy

- Navigate to the **"Route to Web Service and Add SiteMinder Token to Header"** policy in the "Routing" container located in the **"Policies"** module. This can be done for the **"Route to Web Service and Add SiteMinder Token to Message Body"** as well if preferred.
- Drag a **"Trace"** filter from the **"Utility"** filter category and add it to the last filter of the policy so it looks like the screenshot below

- For **"Trace Level"** select **"Debug"** from the drop down list
- Check the **"Include Attributes"** check box.

The routing policy modified with a trace filter.



- Set the Gateway trace level to **"Debug"** by navigating to the **"Services"** module on the left of Policy Studio.
- Right click on the OEG Gateway node and click on **"Settings".**
- Select **"Custom"** and change the trace level to **"Debug"** in the **"Trace level"** drop down field.
- Hit **"F6"** on the keyboard or click on **"Settings"** on the top menu and select **"Deploy".**
- Send a request via explained in section 5.2
- The value of the SiteMinder session token can be seen in bold below.
- This can be used as explained in section 5.2 and 6.2 and inserted either in the HTTP header or message body for testing the policy decision tree.

The result in the Gateway trace console (truncated)

```
DEBUG   13:13:55:418 [1170]     run circuit "Route to Axis
Service and Add SiteMinder Session TokenHeader"...
DEBUG   13:13:55:418 [1170]     run filter [Route to Axis
Service] {
DEBUG   13:13:55:418 [1170]         get connection to host
192.168.0.51 port 7070 scheme http
DEBUG   13:13:55:418 [1170]         new endpoint
192.168.0.51:7070
DEBUG   13:13:55:418 [1170]         Resolved
192.168.0.51:7070 to:
DEBUG   13:13:55:418 [1170]             192.168.0.51:7070
```

```
DEBUG   13:13:55:418 [1170]         connected to
192.168.0.51:7070
DEBUG   13:13:55:418 [1170]         new connection
09B09868,  settings source service-wide defaults
(force 1.0=no, idleTimeout=15000, activeTimeout=30000)
DEBUG   13:13:55:418 [1170]         new client transaction
098B3AD0
DEBUG   13:13:55:418 [1170]         add header
Host:192.168.0.51:7070
DEBUG   13:13:55:418 [1170]         add header Via:1.0
schoemang (Vordel)
DEBUG   13:13:55:418 [1170]         add header
Authorization:Basic Y249aHViZXJ0IGZhcm5zd29ydGgsbz1wb
GFuZXQgZXhwcmVzcyxsPW5ldyB5b3JrLHN0PW55LGM9dXM6Z29vZE5ld3M
=
DEBUG   13:13:55:418 [1170]         add header User-
Agent:Vordel
DEBUG   13:13:55:418 [1170]         new byte counter
DEBUG   13:13:55:433 [1170]         optimise write for
vordelDoc com.vordel.jaxprovider.libxml.Docum
entImpl@6a1140
DEBUG   13:13:55:433 [1170]             ... to non-vordel
output stream
DEBUG   13:13:55:433 [1170]         start client request,
POST "http://192.168.0.51:7070/axis/servic
es/urn:xmltoday-delayed-quotes"
DEBUG   13:13:55:433 [1170]         send prologue: content
length 1550
DEBUG   13:13:55:433 [1170]         add header Content-
Length:1550
DEBUG   13:13:55:433 [1170]         reused connection
09B09868 1 times
DEBUG   13:13:55:433 [1170]         connection keepalive?
0
DEBUG   13:13:55:433 [1170]         add header
Connection:close
DEBUG   13:13:55:433 [1170]         optimise write for
vordelDoc com.vordel.jaxprovider.libxml.Docum
entImpl@6a1140
DEBUG   13:13:55:433 [1170]             ... to vordel
output stream
DEBUG   13:13:55:527 [1170]         HTTP response code
from server: 500
```

```
DEBUG   13:13:55:527 [1170]         add header Set-
Cookie:24
DEBUG   13:13:55:527 [1170]         add header Set-
Cookie2:24
DEBUG   13:13:55:527 [1170]         add header Content-
Type:text/xml; charset=utf-8
DEBUG   13:13:55:527 [1170]         connection processor
made 1 attempts to transact
DEBUG   13:13:55:527 [1170]         handle type text/xml
with factory class com.vordel.mime.XMLBody$
Factory
DEBUG   13:13:55:527 [1170]         add header Via:1.1
schoemang (Vordel)
DEBUG   13:13:55:527 [1170]     } = 1, in 109 milliseconds
DEBUG   13:13:55:527 [1170]     run filter [Add SiteMinder
Session Token into Response Header (Add HTTP Header)] {
DEBUG   13:13:55:527 [1170]     } = 1, in 0 milliseconds
DEBUG   13:13:55:527 [1170]     run filter [Trace] {
DEBUG   13:13:55:527 [1170]             Trace {
java.lang.String
DEBUG   13:13:55:527 [1170]                 }
DEBUG   13:13:55:527 [1170]
attribute.subject.id {
DEBUG   13:13:55:527 [1170]                     Value:
cn=Hubert Farnsworth,o=Planet Express,l=New York,st=NY,
c=us
DEBUG   13:13:55:527 [1170]                     Type:
java.lang.String
DEBUG   13:13:55:621 [1170]                     Type:
netegrity.siteminder.javaagent.ResourceContextDef
DEBUG   13:13:55:621 [1170]                 }
DEBUG   13:13:55:621 [1170]
siteminder.session {
DEBUG   13:13:55:621 [1170]                     Value:
AvBDZ5sAPWafoTAEgvQuz0It9QN7QFmmfXUzn9KcU
NgzMIcb4XfER+vu18CZJTD142JZQiANYNpgZ2dz+x6tW3fvdtm0eZOY/Rs
7WwUzhCX6RoSpgZulFUHKlxRaj6UrTwDrw0QbjN+u+47PqyvRfElhwKE6c
QiVSSeXgv7vyQwuQ2oMJyKwJwjOhhU7magSxYcgt/rse1sU4HdoybFzJf4
PJZ3yUKuo9Rgdk9SUeRArpFRfm2sx2bElLFs0LgfX9kjIRI5p6UPIXuMh/
14IFtdrWxiTsuiaf9EINHi2Et+35lsgW4c/V9jz6Hcw66jai++GnUZiObm
uiJoIYf1/KlcIZPySsf42wVOl/PkbDqT97xS+DKuRSeusESdP1TKLoShTY
I1imCsyUZG/0aew3uWxZQpW/AB1wv54wKczySxJeoRan5a/5nXJnPzklBH
Q/wzFO2msh16S8Nwc3S9/eRW2vRvJS0dspqZ5zyskmRpTC3DpU18WVLSaZ
```

**q0ia2ioK9rPZVl7Ag0kD0y/rKMclRGOfKj8v2HMmAsF4qPfKO8S7CX2uyc
dUzkB6cZaWZNqN92oNRJCri0QrGVOHeOtKQm7P86NR7bdTielGiEtH2FNB
PC2zVu1hb8S4QOECJqfLktJ1uRLLwG1gZmZc9A4TwG5GGDVadn1hzDqjxQ
AZvhrRl+V3iYMCSiGMaajfGCNDZLDrKLLaCA8EKlx6oSlEqlKLhSS44GBF
l9F4nH34RoS67h7Vr3UTBkeCY8w5/I6rsqxyVDtzqtXAgedsrr16ss4HeE
tHi4fzqCOAyCeLyZHk+JFp3FTBkAfjkMh68deGTguBphHg0Qr+ES3E44eG
SuKWTgMAgFzGa4MT4rDZioeI3jxOZ1ByfTGQWlh6DPYrgl+NJDEbiy1Hdh
mIIA4Rzonk34wzIm2h3xKEDHiuzS16I7KtTsWfJl0HjBz/Jc+Uu/0d84Om
1M4lJhvnKcuFP0/ZkVbwxcHrT/oELHJ9tHnw5DsQthl+ufi/II1Bwd8r89
1ZXhlq8pTiQutZVGJZ+3VVUbXfwwAIcJuDcYd3zgz8HOPAwmUtL6f5ItdA
Mq5GTMwzYiCcr5xtKYmd35YGJm4nM6V7CUY**
DEBUG 13:13:55:636 [1170] Type:
java.lang.String
DEBUG 13:13:55:636 [1170] }
DEBUG 13:13:55:636 [1170] }
DEBUG 13:13:55:636 [1170] } = 1, in 109 milliseconds
DEBUG 13:13:55:636 [1170] ..."Route to Axis Service
and Add SiteMinder Session Token Header" complete.

Oracle Enterprise Gateway
May 2011
Author:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment

SOFTWARE. HARDWARE. COMPLETE.