

[OpenAz] Version 86 - end to end integration w sunxacml

Rich.Levinson rich.levinson@oracle.com

Wed Mar 31 12:30:04 EDT 2010

- Previous message: [\[OpenAz\] OpenAz Conference Call - Thursday April 1, 1 PM ET](#)
- Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

To OpenAz maillist:

I have uploaded a "fully integrated" sunxacml release as version 86. There is still cleanup to do, but I wanted to update everyone on the current status.

Attached, for easy access for those interested, are a rudimentary policy generation tool and the generated policy that was used for the integration. The "tool", TestAzApiPolicies.java (really it is more of an outline that could evolve to a tool), is an adaptation from the sample sunxacml module: SamplePolicyBuilder.java. This turned out to be quite a flexible tool that appears poised to be able to generate xacml policies with a simple syntax, that I think has some interesting possibilities, esp wrt simplification of the concepts in xacml, and how xacml in this syntax can be related to other commonly used Java policy models. The tool is still in a very primitive state, but it should be fairly straight-forward to follow. The basic model is built around "attribute expressions", where I created a simple class AttributeMatchExpression.java (also attached), that contains the metadata for matching attributes as is found in the xacml Targets. The result is a sequence of simple attr expressions that identify the xacml attr id, the xacml attr value used on the policy side to compare against the attr value rcvd in a request, a xacml match-id indicating the type of comparison, as well as other attribute metadata such as the sunxacml designator type which corresponds generally to the AzApi AzCategoryId, the attribute issuer, and the xacml data type.

The attached policy was generated by the tool and works with the existing test programs: TestAzApi, TestStyles, and BulkDecideTest, as well as some of the sunxacml sample code.

So, what happens now is when any of the test programs are run (assuming they are called with the correct arguments that cause the sunxacml pdp to be used as the AzService, thru the SimpleConcreteSunXacmlService.java module from the openaz/pdp-org.openliberty.openaz.pdp.provider project-package) described in previous emails), the Requests they generate actually are directed to the sunxacml pdp which evaluates the request against the attached policy (TestAzApi-GeneratedPolicy.xml), and returns the results including obligations that are also defined in the policy by the tool.

We can discuss it further on Thursday at the OpenAz conf call so that everyone can understand the details. However, most of the new information is contained in the attached files, TestAzApiPolicies.java and AttributeMatchExpression.java (from the openaz/test-test.policies project-package).

The next step is to clean up the methods to run these tests a bit so people will be able to download, build, and run w ant. This can be done now, in principle, at least the build part (still just run ant from the openaz/test/build directory), but the run needs some work.

Thanks,
Rich

----- next part -----

A non-text attachment was scrubbed...

Name: TestAzApi-GeneratedPolicy.xml

Type: text/xml

Size: 11154 bytes

Desc: not available

Url : <http://lists.openliberty.org/pipermail/openaz/attachments/20100331/6694513a/attachment-0001.xml>
----- next part -----
An embedded and charset-unspecified text was scrubbed...
Name: TestAzApiPolicies.java
Url: <http://lists.openliberty.org/pipermail/openaz/attachments/20100331/6694513a/attachment-0002.pl>
----- next part -----
An embedded and charset-unspecified text was scrubbed...
Name: AttributeMatchExpression.java
Url: <http://lists.openliberty.org/pipermail/openaz/attachments/20100331/6694513a/attachment-0003.pl>

- Previous message: [\[OpenAz\] OpenAz Conference Call - Thursday April 1, 1 PM ET](#)
 - Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)
-

[More information about the OpenAz mailing list](#)