[OpenAz] Version 97: Easy XACML syntax with OpenAzPolicyReader

Rich.Levinson rich.levinson at oracle.com
Thu Jul 15 00:43:36 EDT 2010

- Previous message: [OpenAz] [Fwd: Today's conference called is cancelled - next call Thu Jul 22, 2010 1PM ET]
- Next message: [OpenAz] Version 98: Preliminary skeleton tutorial w OpenAzPolicyReader
- **Messages sorted by:** [date] [thread] [subject] [author]

```
To OpenAz maillist:

Version 97 of OpenAz is now available for use. This is a preliminary
version that essentially includes the full runtime functionality (but
not yet the tutorial materials, and the download and build, while
theoretically should work by running ant in the test/build directory,
still has not been tested outside my own environment, but will be soon)
described in earlier email:
http://lists.openliberty.org/pipermail/openaz/2010-June/000069.html

The main new thing in version 97 is the "easy to use XACML abbreviated
language", where "abbreviated" means that the syntax has been
abbreviated to remove the XML artifacts, and all XACML identifiers have
abbreviations, including element and attribute names, as well as
selected values for those attributes.

An example of this abbreviated syntax was described in above mentioned
email, and a current version is attached to this message,
TestAzApi-OpenAz-Pseudo-Policy.txt, and may also be found in OpenAz
directory: openaz/test/policy.

The program that can process this syntax is named OpenAzPolicyReader,
and it may be found in the openaz/test/src/test/policies directory.

Basically, OpenAzPolicyReaderis a policy interpreter/compiler that takes
the simplified, line at a time syntax representation of xacml, which is
very similar to commonly used line at a time grant syntaxes. In this
case each line is essentially a list of name/value pairs for the main
xacml constructs:

    * PolicySet
    * Policy
    * Rule
    * Target (for each of the above)
    * Condition (simple, no attempt yet at complex syntax)
    * Obligations
    * AttributeMatchExpressions

The latter construct, AttributeMatchExpression, which was introduced in
earlier emails, is the common denominator for use by the other elements.
Basically, Targets, Conditions, and Obligations contain lists of
AttributeMatchExpressions, which are simple comparisons of some
specified attribute with a value in the Match expression. The top 6
elements can be considered as "grouping" the Expressions in a
hierarchical manner where the expressions filter down to the eventual
decision criteria for a particular request. It is fundamentally the same
as common "grant syntaxes", except it is richer in the hierarchical
grouping capabilities, as well as combining capabilities, which enable
some interesting "dialects", some of which have already been
demonstrated in the examples.

In order to test the language, the attached "easy xacml" file was read
by OpenAzPolicyReader and it produced the second file which is attached,
TestAzApi-GeneratedPolicy.xml, which is essentially the same policy file
that has been used for testing earlier OpenAz releases.
```

Note: for "historical context", when the SunXacml sw was integrated with OpenAz earlier this year, and we started developing XACML policies for use in testing AzApi and PepApi:

* the overhead of maintaining the xml policy files quickly became a productivity drag, and so we introduced a utility based on the sunxacml sample program, SamplePolicyBuilder, which was expanded to be XacmlPolicyBuilder, which is where the AttributeMatchExpressions (AMEs) were introduced as an easy way to generate XACML Policies.
* Unfortunately, it soon became apparent that the overhead of maintaining the AMEs in XacmlPolicyBuilder was another productivity drag, and so it was decided to provide a simpler way to manage the AMEs rather than modifying the code in XacmlPolicyBuilder.
* The result of this is the "easy syntax" attached and used by OpenAzPolicyReader.
* It has not escaped notice that even the "easy syntax" may be an issue if these 2 syntaxes of XACML are mixed up in the same environment. It turns out that the easy syntax, itself, that is attached was generated by XacmlPolicyBuilder, and so it does not seem unreasonable that it may be a fairly straight-forward task to modify XacmlPolicyBuilder to take an existing XACML Policy file and use that to generate an easy syntax file. The point is that if people find the easy syntax useful, then there should be a clear path to be able to represent existing XACML policy files in that syntax without much difficulty, however, that is not a current agenda item.

Note also that SunXacml basically supports XACML 2.0 minus a few features that are still TBD as to whether SunXacml needs to be upgraded or some other strategy makes sense. However, the PolicySets should be pretty much XACML 2.0 compatible, except possibly some legacy XACML 1.2 artifacts, such as the default syntax for empty Targets may need to be removed.

The specific syntax of the easy syntax xacml language is "flexible" in that all the tokens are defined in tables in OpenAzPolicyConstants.java, also in the src/test/policies directory. So, if people wanted longer or shorter names for the line names, or any of the name/value pairs, that would be straight-forward. Also, users can define their own abbreviations for any commonly used attribute data or metadata.

The main "philosophy" of the language is that the XACML language in XML format as presented in the XACML specifications is too complicated for general customer use because:

1. It has a lot of xml syntactical structure that is hard to penetrate for people who don't spend a lot of their time using xml.
2. The names of everything in xacml are generally longer than antidisestablishmentarianism, which makes the whole thing basically unapproachable for ordinary users who just want to define policies.

So the "easy" language removes the xml syntax and replaces it with sequential lines, which is generally possible since xml is generally sequential, as in a SAX parser. The only cross-line syntax is that there are certain orders to the lines that are required, but this is straight hierarchical, so it is very intuitive. i.e. start with policysets which contain policies etc. The only "recursive" element is the PolicySet, and this recursion was incorporated using an mlevel token, which is just an integer starting at 0 for the root policy and incremented by one for each parent child relation between policysets. There is also the notion of "terminators" where a sequence is ended by the appearance of a higher construct, which again is intuitive. For example if a policyset with mlevel 3, appears after a prev policyset at mlevel 3 followed by a bunch of stuff, then the prev policy needs to be wrapped up and subsequent "stuff" is associated with the new policyset.

However, the tutorial which we will begin to introduce in the next several days will start with very simple 5-10 line policysets to show how the whole thing works.

```
If one wanted to experiment with the current release, prior to having
some cleaned up instructions, the basic idea is the following:

    * run Java on OpenAzPolicyReader with
      TestAzApi-OpenAz-Pseudo-Policy.txt as a command line argument.
    * there will be a gigantic output of debugging information, followed
      by the generated file, which should be identical to
      TestAzApi-GeneratedPolicy.xml, which is also attached.
    * this generated file may then be used as input to running any of
      the other test programs: TestStyles, QueryTest, BulkDecide, and
      TestAzApi, all of which have their test cases supported by the
      attached xml policy file, and by the "pseudo" policy file.
      Essentially, all the tests are now runnable based on the easy
      syntax xml file attached.

As indicated, additional materials to help introduce users to the
operational capabilities will be uploaded to the site in the coming days
and weeks. The site is essentially at full functionality now, but is
obviously still in very skeletal condition, however, conceptually, at
this point all the essentials of AzApi are included.

These materials will be subject for discussion at next weeks scheduled
call: Thu July 22, 2010 at 1PM EDT, which will have an agenda posted
next week.

    Thanks,
    Rich

-------------- next part --------------
An HTML attachment was scrubbed...
URL: http://lists.openliberty.org/pipermail/openaz/attachments/20100715/3da3a31a/attachment-0001.html
-------------- next part --------------
An embedded and charset-unspecified text was scrubbed...
Name: TestAzApi-OpenAz-Pseudo-Policy.txt
Url: http://lists.openliberty.org/pipermail/openaz/attachments/20100715/3da3a31a/attachment-0001.txt
-------------- next part --------------
A non-text attachment was scrubbed...
Name: TestAzApi-GeneratedPolicy.xml
Type: text/xml
Size: 36363 bytes
Desc: not available
Url : http://lists.openliberty.org/pipermail/openaz/attachments/20100715/3da3a31a/attachment-0001.xml
```

- Previous message: [OpenAz] [Fwd: Today's conference called is cancelled - next call Thu Jul 22, 2010 1PM ET]
- Next message: [OpenAz] Version 98: Preliminary skeleton tutorial w OpenAzPolicyReader
- **Messages sorted by:** [date] [thread] [subject] [author]
