Supporting XACML authorization policy authoring

Piotr Bała
Krzysztof Benedyczak
Tomasz Królikowski

UNICORE Summit 2011

What is XACML?

- “eXtensible Access Control Markup Language”
- Language for describing authorization policies in XML.
- WHO can have access to WHAT, under which CONDITIONS, for which PURPOSES.
- OASIS Standard.
- Very complicated syntax, strong expressibility, machine processable.
- Many implementations (both Open Source and proprietary).

Use of XACML

- UNICORE -The XACML policy describes how and by whom, actions on WSRF resources and plain Web Services can be performed.
  - Policy is rarely modified in standard deployments but it sometimes happens.
- Fedora Repository - the XACML language is used to control access to Fedora resources.
- PayPal – the XACML centralizes and streamlines access control in applications used to service a global customer base.
- Many others (e.g. IBM in Tivioli).

Problems with XACML

- XACML is inherently complicated!
  - The default UNICORE policy has ca 200 very long and complex lines.
- It is very difficult to edit the policy.
- One of the approaches is to use a simplified language which is then translated to XACML
  - The Argus approach,
  - However this limits policy features!

```
  <Rule RuleId="Permit:TargetSystemFactoryService_for_user" Effect="Permit">
    <Description> Full access to the TargetSystemFactoryService is granted for authorised users. </Description>
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">TargetSystemFactoryService</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#anyURI"
MustBePresent="true"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <Condition>
    <Apply  FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
        <SubjectAttributeDesignator AttributeId="role" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
      </Apply>
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">user</AttributeValue>
     </Apply>
    </Condition>
  </Rule>

<Rule RuleId="Permit:Default_SMS_for_user" Effect="Permit">
    <Description> Access to the default_storage service is granted for authorized users </Description>
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">default_storage</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:unicore:wsresource" DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">StorageManagement</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#anyURI"
MustBePresent="true"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <Condition>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
        <SubjectAttributeDesignator AttributeId="role" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
      </Apply>
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">user</AttributeValue>
      </Apply>
```

etXACML

- etXACML = Web-Based Tool + Console Tool
- The aim: **comprehensive set of tools for supporting XACML policies authoring**.
- Java Technology + HerasfAF Core XACML
- Features:
  - Validation of the the syntax based on the XACML 2.0 Policy and Context schema.
  - Evaluation engine can check all the rules in an XACML Policy before making the final decision for a request.

etXACML

- Debugger
  - Extended evaluation, Policy is broken down into parts, and for each individual a response is computed. We get an additional information which elements were involved in the evaluation and which rules had an impact on the final decision.
- UNICORE Test
  - Special set of requests with answers which verify if the tested policy has the standard properties of the default UNICORE policy (admin access, user access, ...)
- Meta-Tests
  - For each item of a policy, lets you to select the expected result of the evaluation and then compare it with the actual, received result.