OpenSSO QA Test Automation

XACML Test development and execution

Prepared by Sridhar Enugula

1. Introduction

This document describes the archetectural/implementation details primarily focusing on the automation of the XACML feature. This document describes the following:

- Requirements
- How tests are organized
- Execution Details
- Tests/Features in framework
- Interpreting the report
- Debugging the test failures
- How to add new tests

2.Requirements

The primary requirements are:

- XACML automated testing assumes that the user is familiar with the QA automation framewrok (please read "Overall Architecture of the Automation Framework" document for more details about the framework and workspace)
- Basic knowledge about the Federated Access Manager 8.0 and XACML
- Federated Access Manager 8.0 war is deployed/configured on the supported container**.**

3.How Tests are Organized

1. XACML is a new feature in FAM8.0 an implementation of XACML 2.0 OASIS standard specification , FAM8.0 supports XACMLDecisionQuery and XACMLAuthZDecisionStatement specified by SAML2.0 profile of XACML2.0 in FAM8.0 .The following test cases (decision statements) are being automated

- Permit
- Deny
- Indeterminent

The above test cases are for the POST and GET actions.

Test Organization details:

1. Directory and file structure for authentication module features is laid out as below :

- The above listed features are packaged under the "xacml"package.
  <TEST_HOME>/source/com/sun/identity/qatest/xacml
- The utility/common classes authentication module are at
  <**TEST_HOME>/source/com/sun/identity/qatest/common/**
- All properties and configuration files for each feature are at
  <**TEST_HOME>/resources/xacml**
- Runtime classes and xml files will be genereated under
  <**TEST_HOME>/<TEST_SERVER1>/built/classes**

4. Execution details

4.1 How to execute tests :

This section describes how to execute authentication tests.

1. Deploy Federated Access Manager 8.0 and configure
2. Make sure you have the Configurator-<hostname>.properties that corresponds to the test server on which the tests are run.Please read, "Overall Architecture of the Automation Framework" document for more details.
3. For XACML module tests to execute properly, XACML featur need to be configured properly for the tests to run., Modify the following properties files:
   <**TEST_HOME>/resources/xacml/xacmlPolicyGlobal.properties**
   <**TEST_HOME>/resources/xacml/xacmlPolicyTest.properties**
   <**TEST_HOME>/resources/xacml/XACMLTest.properties**

   NOTE : xacmlPolicyGlobal.properties and xacmlPolicyTest.properties doesnot need to be modified unless if you would like create your own policies, read the comments in these property files for more details. XACMLTest.properties contains the name of PEP and PDP and meta alias details for generating and loading metadata for XACML PEP and PDP

IMPORTANT NOTE BEFORE RUNNING THE TESTS :
When running the xacml module make the following changes in AMClient.properties
com.sun.identity.agents.app.username=amadmin
com.iplanet.am.service.password=<$amadmin_password>

4.2 The Exection details :

1. Run the following command to execute xacml module:
   ant -DERSER_NAME1=<test_host_name> module
2. Each test feature is executed based on the feature related properties file.The properties for each test need to modified appropriatly based on the exepected output and the appropriate parameters . (For more details refer to Test Details section of this document)
3. When each test/feature is run, first the required setup for that test is done, this is done in the setup method of the feature related java class.This is governed by BeforeClass annotation.This is basically prepares system for execution of the tests.

4. The @Test annotation in the java source files represents each test.Each tests can be run at random unless there is a Dependson annotation for that test or class.
5. @Afterclass annotation represents what should be done finally,Most cases this where the cleanup is done after system has done executing the tests.

5. Test Details :

5.1 Current Test Features/Classes/Data/Property files :

Test Class/Feature	Related PropertiesFiles	Description
XACMLTest.java XACMLClient.java	XACMLTest.properties xacmlPolicyGlobal.properties xacmlPolicyTest.properties	Please read #1 below

- **# 1 XACMLTests :**
  Before running the tests , the following need to be properly configured
  - **XACMLTest.properties:** This is file contains xacml configuration data and the values of each of xacml module specific data should be properly configured or creation of each module.This is located at <qatest>/xml/xacml
  - **xacmlPolicyGlobal.properties:** This is the policy related properties data file. This file is located under <qatest>/resources/xacml .This file is read by the PolicyCommon class for creating the policies that are needed for xacml
  - **xacmlPolicyTest.properties:** This is the policy related properties data file. This file is located under <qatest>/resources/xacml .This file is read by the PolicyCommon class for creating the policies that are needed for xacml
  - **XACMLClient.java** is the XACMl client which is called to create the XACML(SOAP) request and get the response.
  - **Adding Tests Cases:** To add new Test cases, If erequired additional configuarion add the configuration details in the XACMLTest.Properties and if any other corresponding policy related into the Policy files mentioned above.

6. Interpreting Authentication Automated Test Results

After execution of xacml test module, the results will be located under <REPORT_DIR>/<TEST_HOST_NAME>/<EXECUTION_MODE>/EXECUTION_DATE_TIME> , browse index.html in this directory.The overall results will be displayed. Click on the link for <EXECUTION_MODE>-authentication fopr the detailed test report for that execution group, Evaluate the number of tests passed/failed/skipped.More details on reading the test results , please refer to overall automation framework document.

7. Debugging the automated tests

Debugging the failures is very critical thing in automation, Here few ways to debug the xacml failures:

- Check the test execution log, The detailed log for each test case can be found under report directory.<REPORT_DIR>/<TEST_HOST_NAME>/<EXECUTION_MODE>/<EXECUTION_DATE_TIME>/logs.

- Check the Server Logs and find the root cause of the failures
- Execute the test case manually to differentiate if it is a product or test bug or change in the product that causing the failure.