pam_xacml developer

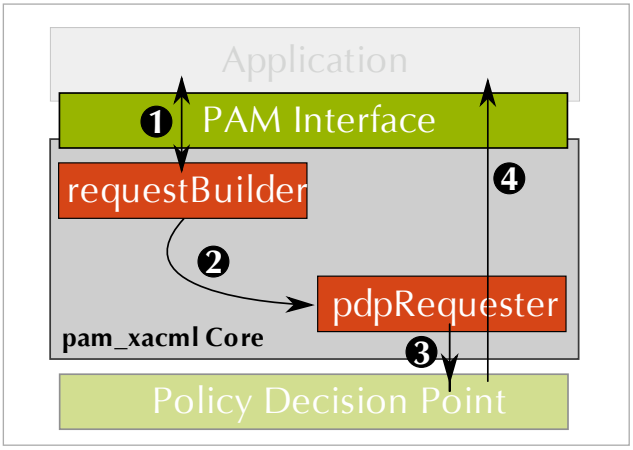
Contact: pamxacml-developer@lists.sourceforge.net
Authors: Andreas Klenk, Tobias Heide
Documentation: Tobias Heide
Supported by: Universität Tübingen and Technische Universität München

1.Preface

pam_xacml was designed to be extendible by developers. It follows an object oriented approach implemented in C++. This document will describe the architecture of pam_xacml to give you the ability to implement enhancements to it's functionality.

2.Overall Design

pam_xacml consists of three blocks. The *core* controls the submodules which handle the communication between application and PDP.



To extend pam_xacml there normally is no need to touch the core, but only the requestBuilders or the pdpRequesters. These two submodule will be described in the following sections.

requestBuilder

The requestBuilder has the responsibility to create the XACML request that is sent to the PDP by pam_xacml. All of the request builders inherit from `src/core/requestBuilders/AbstractRequestBuilder.h`. This superclass only defines one function called buildAuthzRequest() which returns a string containing a full (valid) XML document for the XACML request.

Provided with the distribution are several builders:

- ApplicationAsRequestBuilder: the application is asked for the XACML request over the PAM conversation function.
- PAMRequestBuilder[1]: pam_xacml creates a request according to a XML-template that has to be provided by the user.
- ScriptRequestbuilder: pam_xacml calls an external script which in turn has to provide the XACML request.

pdpRequesters

PDP requesters take the XACML request and send them to a PDP. When they receive an answer, it is returned to pam_xacml. All PDP requesters inherit from `src/core/pdpRequesters/pdpRequesterFactory.h` which defines an abstract class PDPRequester. This class has one method which takes a string as argument (the XACML request) and returns a string containing the full XACML-Decision.

Provided with the distribution are several requesters:

- BesterPdpRequester: This class is able to send the request to the XACML implementation by Joseph Bester.
- hannesPdpGsoapRequester: (currently dysfunctional): For internal use at the Whilhem Schickard Institute.
- pdpFileRequester: returns the contents of a file as static response to any XACML request.
- PDPSimpleRequester: for use with the SimplePDP provided with pam_xacml (see `PDP/SimplePDP`) which is based upon the SUN XACML implementation.

3.Test Environment

Some tests are provided with pam_xacml. They are located in the subdirectory tests/ of the distribution.

[1] This requestBuilder is called INTERNAL in the pam.d configuration file.

Script Files

Some scripts are provided to test the script based XACML request generation and parsing of obligations. These scripts are called requestBuilderScript.pl and obligations-receiver.sh, repsectively. They are very simple and just demonstrate how things work generally.

Unit Tests

- PAMRequestBuilderTest: tests the internal request builder of pam_xacml
- xacmlLibraryTest: tests the XACML library which is currently used to build requests for the PDP by Joseph Bester.

General Tests

- pamxacml_test: This script acts like an application using pam_xacml for authorization. A complete call through the PAM library is done, and the test script even understands obligations. Please note, that this script will need some further configuration. A file under `/etc/pam.d` named pamxacml_test shall be created which contains a call to pam_xacml.so in it's module chain. The rest of the configuration is up to you, whatever you want to be tested...
- standalone_test: This test is currently excluded from the build. It was used to provide testing facilities without using the PAM library by simulating the functions of the PAM library. Have a look at the pam_facade files to get an impression how this works.

4.Performance Measurement Data

The libtiming module located unter src/core was used to gain several performance data when using pam_xacml. This library is very basic and does not do anything special but logging timestamps at several points in the control flow. The source should be self-understanding.

5.Further Information

Further information can be obtained from the user manual provided with pam_xacml.