Authentication and User Attributes

Introduction

This document collects technical notes on how to supply user attributes for evaluation of Fedora's XACML authorization policies.

For a practical introduction to using Fedora's servlet filters for authentication and user attributes, see Securing Your Fedora Repository. The current document is a technical appendix to that section.

Injecting Arbitrary User Attributes

Fedora 2.1.1 provided a mechanism to include user attributes from arbitrary source(s), and merge them with attributes provided by Tomcat realms or login modules. Fedora 2.2 continues this with its servlet filter approach, independent of specific servlet container. The Fedora code looks for a request attribute whose name equals the string constant fedora.server.Context.FEDORA_AUX_SUBJECT_ATTRIBUTES. A request attribute found under that name is taken as a Map, mapping names to values, and so giving additional user subject attributes. Currently, name must be a String and this is unlikely to change. Value must also be a String, and later this might be relaxed to include String[], to allow attributes with multiple values. Other types of value are not serviced. The effect within Fedora of having a key => value pair "a" => "b" in the Map is the same as the current user having an attribute named "a" with value "b", e.g., as defined in fedora-users.xml.

An arbitrary site-supplied servlet filter must create the map and populate it, and put it into the http servlet request as attribute named fedora.server.Context.FEDORA_AUX_SUBJECT_ATTRIBUTES. Fedora will then look for it, and use the attributes in xacml authorization. This services any subject attribute source, and remains source-neutral, i.e., you could use it to implement Shibboleth, but it doesn't favor Shibboleth.

If doing this, it would be best to isolate your interface code to be independent both of being in a servlet filter and also of interacting with the Map. This is so you could later refactor the code more in line with Fedora 2.2 servlet filters.

This means of introducing arbitrary attributes has been tested by another developer successfully before Fedora 2.2, but not yet with Fedora 2.2. Eventually, the Fedora servlet filter approach will be documented as a best practice to providing user attributes.