Authorization Delegates

Overview

Fedora Authorization Delegates allow you to implement one interface to enforce access control over your Fedora repository. This interface, FedoraAuthorizationDelegate, has callbacks that allow you to restrict ModeShape operations and filter search results. After following these configuration steps, Fedora's REST endpoints will respond with 403 response codes when the requested action is unauthorized by the authorization delegate.

Use of an authorization delegate and Fedora-specific authorization is optional. You can also configure Fedora to run without API security. You may want to only enforce container authentication or leave the service running completely unsecured, behind a firewall for instance. For details, see How to configure Fedora without authorization.

Fedora Administrators (fedoraAdmin user role)

The authorization delegate is not consulted when servlet credentials identify a client with the **fedoraAdmin** role. When the container has authenticated the connected client as a **fedoraAdmin**, all actions are permitted and PEP is bypassed.

FedoraAuthorizationDelegate Implementations

There are two reference implementations available:

- Basic Role-based Authorization Delegate - An authorization delegate that operates on three fixed roles that may be assigned throughout the repository tree. (reader, writer, admin)
- XACML Authorization Delegate

You can also create an authorization delegate implementation and perform security checks differently, possibly including calls to remote services.

Two files contain the configuration options for authorization delegates:

- **repo.xml**: the global repository configuration file. Three beans enable the PEP extension:
  - *modeshapeRepoFactory*: should contain a dependency on the *authenticationProvider* bean
  - *authenticationProvider*: should specify the *ServletContainerAuthenticationProvider* class, so that the servlet container handles authentication
    - This bean should have a property "fad" that points to the *fad* bean, to enable the servlet container authentication provider to use the authorization delegate
  - *fad*: should point to your class with the authorization delegate implementation
- **repository.json**: the ModeShape configuration file. It contains a security section, where the **internal session** authentication between Fedora and the ModeShape storage layer is configured. Note that the roles configured here do not apply to end user authentication and authorization..

Step-by-step:

1. Open the repo.xml file in your Fedora web application.
2. Add your authorization delegate implementation as a bean in this file and give it the ID of "fad". Your authorization delegate bean may include more specific configuration details than the example.
3. Now add the Fedora ModeShape Authentication Provider bean. (see repo.xml example)
4. Make sure that your modeshapeRepofactory bean has the depends-on attribute pointing at the authenticationProvider (see repo.xml example).
5. Open your repository.json file.
6. Add org.fcrepo.auth.ServletContainerAuthenticationProvider as a provider in the security section. (see repository.json example)

Example repo.xml (repository and security beans)

```
<bean name="modeshapeRepofactory" class="org.fcrepo.kernel.spring.ModeShapeRepositoryFactoryBean"
  depends-on="authenticationProvider">
    <property name="repositoryConfiguration" value="${fcrepo.modeshape.configuration:repository.json}" />
</bean>

<bean name="fad" class="your.own.implementation"/>

<bean name="authenticationProvider" class="org.fcrepo.auth.ServletContainerAuthenticationProvider">
  <property name="fad" ref="fad"/>
</bean>
```

Example repository.json (security section)

```
"security" : {
  "anonymous" : {
    "roles" : ["readonly","readwrite","admin"],
    "useOnFailedLogin" : false
  },
  "providers" : [
    { "classname" : "org.fcrepo.auth.ServletContainerAuthenticationProvider" }
  ]
},
```