Principal Providers

Fedora Principal Providers allow a Fedora repository to pull in user security and role designations from other sources (e.g. LDAP).

Principal Providers are implemented as servlet filters that are added to the Shiro filter chain between the initial authentication filter (ServletContainerAuthFilter) and the final authorization filter (WebACFilter).

Different derivatives of the PrincipalProvider class can be initialized differently, either through the repository.json file, other credential files, from information sent via HTTP header or by connecting to external information sources such as LDAP.

Configuration

Principal providers are configured in Fedora's Spring configuration by doing the following:

1. Add a `<bean>` definition for the desired provider, including any necessary configuration parameters. See below for the configuration parameters for the providers that exist in Fedora's core codebase.
2. Add the name of the bean to the `filterChainDefinitions` line in the configuration of the `org.apache.shiro.spring.web.ShiroFilterFactoryBean`. The relevant line starts with `/**`, which means "filter all requests". What follows is a comma-separated list of filter bean names. The request proceeds through the filters from left to right.

Here is the complete default Spring filter configuration used by the fcrepo-webapp:

Spring configuration of principal providers as filters

```
<!-- Authentication Filter -->
<bean id="servletContainerAuthFilter" class="org.fcrepo.auth.common.ServletContainerAuthFilter"/>

<!-- Principal Provider Filter: Delegate Header -->
<bean name="delegatedPrincipalProvider" class="org.fcrepo.auth.common.DelegateHeaderPrincipalProvider"/>

<!-- Authorization Filter -->
<bean id="webACFilter" class="org.fcrepo.auth.webac.WebACFilter"/>

<!-- connect the filters into a chain -->
<bean id="shiroFilter" class="org.apache.shiro.spring.web.ShiroFilterFactoryBean">
  <property name="securityManager" ref="securityManager"/>
  <property name="filterChainDefinitions">
    <value>
      <!-- The Auth filter should come first, followed by 0 or more of the principal provider filters, -->
      <!-- and finally the webACFilter -->
      /** = servletContainerAuthFilter,delegatedPrincipalProvider,webACFilter
    </value>
  </property>
</bean>
```

Classes

Container Roles Principal Provider

ContainerRolesPrincipalProvider is a PrincpalProivder that obtains its set of principals from web.xml.

```
<bean name="containerRolesProvider" class="org.fcrepo.auth.common.ContainerRolesPrincipalProvider">
  <property name="roleNames">
    <util:set set-class="java.util.HashSet">
      <value>tomcat-role-1</value>
      <value>tomcat-role-2</value>
    </util:set>
  </property>
</bean>
```

New roles must be specified in web.xml as shown below.

```
<auth-constraint>
  <role-name>fedoraUser</role-name>
  <role-name>fedoraAdmin</role-name>
  <role-name>tomcat-role-1</role-name>
  <role-name>tomcat-role-2</role-name>
</auth-constraint>
```

Please refer to the servlet container authentication document for further configuration details.

HTTP Header Principal Provider

HttpHeaderPrincipalProvider is a Principal Provider that obtains its initial set of principals from HTTP header requests.

```
<!-- Optional PrincipalProvider that will inspect the request header, "some-header", for user role values -->
<bean name="headerProvider" class="org.fcrepo.auth.common.HttpHeaderPrincipalProvider">
  <property name="headerName" value="some-header"/>
  <property name="separator" value=","/>
</bean>
```

Delegate Header Principal Provider

DelegateHeaderPrincipalProvider is a Principal Provider that uses the `On-Behalf-Of` HTTP header to switch the user principal to the principal given in the header. This switch is only performed if the authenticated user has the fedoraAdmin container role.

```
<bean name="delegatedPrincipalProvider" class="org.fcrepo.auth.common.DelegateHeaderPrincipalProvider"/>
```