How to Configure Servlet Container Authentication

Fedora 4 uses servlet container authentication (Realms) to provide minimal protection for your repository, including the set up of "superuser" accounts. User credentials are configured in your web application container, usually in a properties file or XML file. By configuring superuser accounts you can require authentication for all management (write) operations. This document describes how to set up Fedora and either Tomcat or Jetty to enable HTTP Basic Authentication, using simple user files. Consult your web application server documentation for other ways to configure and manage users; Fedora can handle any user principal passed to it by the servlet container, as provisioned by any of the container's supported authentication mechanisms.

The superuser role is **fedoraAdmin**. This is comparable to the **fedoraAdmin** superuser role in Fedora 3, used for Fedora 3 API-M operations.

- Configure your repo.xml file
- Configure your repository.json file
- Configure your web.xml
- Configure your web application container
  - Jetty
  - Tomcat

The Fedora authorization modules reside in separate source code modules from the core Fedora web-application.

- WebAC : https://github.com/fcrepo4/fcrepo-module-auth-webac
- RBACL : https://github.com/fcrepo4/fcrepo-module-auth-rbacl
- XACML : https://github.com/fcrepo4/fcrepo-module-auth-xacml

As a result, each release includes pre-built Fedora "webapp-plus" war files that have the authorization modules included. You are recommended to use one of these "webapp-plus" war files as a starting point for having an authorization-enabled deployment.

- Webapp-plus releases: https://github.com/fcrepo4-exts/fcrepo-webapp-plus/releases

You can then follow the guidelines in the Best Practices - Fedora Configuration document to specify site-specific "repo.xml" and "repository.json" configurations, as further described below.

1. ## Configure your repo.xml file

   Add the beans *authenticationProvider* and *fad* to your repo.xml file, and make the *modeshapeRepofactory* bean dependent on *authenticationProvider*. Use the class **org.fcrepo.auth.ServletContainerAuthenticationProvider** as your authentication provider. Here is an example repo.xml that configures authentication and authorization using the Basic Roles authorization delegate.

   To specify a local repo.xml configuration, provide the system property as follows:

   ```
   JAVA_OPTS="... -Dfcrepo.spring.repo.configuration=file:/local/repo.xml"
   ```

repo.xml with authentication configured

```
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:util="http://www.springframework.org/schema/util"
    xsi:schemaLocation="
    http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-
3.0.xsd
    http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-
context-3.0.xsd
    http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.
xsd">

    <!-- Context that supports the actual ModeShape JCR itself -->

    <context:annotation-config/>

    <bean name="modeshapeRepofactory"
        class="org.fcrepo.kernel.impl.spring.ModeShapeRepositoryFactoryBean"
        p:repositoryConfiguration="${fcrepo.modeshape.configuration:classpath:/config/servlet-auth
/repository.json}"
        depends-on="authenticationProvider"/>

    <bean class="org.modeshape.jcr.JcrRepositoryFactory"ModeShapeEngine" init-method="start"/>

    <bean id="connectionManager" class="org.apache.http.impl.conn.PoolingHttpClientConnectionManager" />

    <!-- 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>

    <util:set id="principalProviderSet">
        <ref bean="headerProvider"/>
    </util:set>

    <bean name="fad" class="org.fcrepo.auth.roles.basic.BasicRolesAuthorizationDelegate"/>

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

    <!-- For the time being, load annotation config here too -->
    <bean class="org.fcrepo.metrics.MetricsConfig"/>
</beans>
```

2. ## Configure your repository.json file

Modify the security section to enable both authenticated (via authentication provider) and internal sessions between Fedora and ModeShape.

To specify a local repository.json configuration, provide the system property as follows:

```
JAVA_OPTS="... -Dfcrepo.modeshape.configuration=file:/local/repository.json"
```

It should contain a "security" element that matches this block:

repository.json security

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

3. ## Configure your web.xml

Configure your **web.xml**.

Modify fcrepo-webapp/src/main/webapp/WEB-INF/web.xml by uncommenting the security configuration

```
  <!--Uncomment section below to enable Basic-Authentication-->
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>Fedora4</web-resource-name>
      <url-pattern>/*</url-pattern>
      <http-method>DELETE</http-method>
      <http-method>PUT</http-method>
      <http-method>HEAD</http-method>
      <http-method>OPTIONS</http-method>
      <http-method>PATCH</http-method>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
      <role-name>fedoraUser</role-name>
      <role-name>fedoraAdmin</role-name>
    </auth-constraint>
    <user-data-constraint>
      <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
  </security-constraint>
  <login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>fcrepo</realm-name>
  </login-config>
```

The "auth-constraint" element must contain the roles defined as your users (see below for jetty and tomcat).

4. ## Configure your web application container

- **Jetty**
  - Create your **jetty-users.properties** file. This file contains entries in the format *username: password [, role, ...]*, where
    - *username* is the user's login id (the principal)
    - *password* is the user's password
    - *role* is the servlet role they are assigned upon login; jetty allows you to specify any number of roles (or no role at all). Fedora currently supports two roles: **fedoraAdmin**, which is the superuser role, and has rights to do everything; and **fedoraUser**, which is a user role, and must be granted permissions by the Policy Enforcement Point to perform actions.

    Sample **jetty-users.properties** file that contains three users, two of whom are regular users, and the third of whom (fedoraAdmin) is a Fedora superuser:

    jetty-users.properties

    ```
    testuser: password1,fedoraUser
    adminuser: password2,fedoraUser
    fedoraAdmin: secret3,fedoraAdmin
    ```

- Configure your Jetty login realm.
  - Standalone
    Modify your **jetty.xml** file to configure the login realm and include the jetty-users.properties file:

    jetty.xml login service

    ```
    <Configure class="org.eclipse.jetty.webapp.WebAppContext">

      <!-- Set this to the webapp root of your Fedora 4 repository -->
      <Set name="contextPath">/</Set>
      <!-- Set this to the path of of fcrepo4 WAR file -->
      <Set name="war"><SystemProperty name="jetty.home" default="."/>/webapps/fcrepo4</Set>

      <Get name="securityHandler">
        <Set name="loginService">
          <New class="org.eclipse.jetty.security.HashLoginService">
            <Set name="name">fcrepo4</Set>
            <!-- Set this to the path to your jetty-users.properties file -->
            <Set name="config"><SystemProperty name="jetty.home" default="."/>/path/to
    /jetty-users.properties</Set>
          </New>
        </Set>
      </Get>

    </Configure>
    ```

  - Embedded in Maven
    - The fcrepo-webapp Maven project includes jetty-maven-plugin. The property *jetty.users.file* sets the location of the **jetty-users.properties** file. Run the fcrepo-webapp server with the following system property:

      ```
      -Djetty.users.file=/path/to/jetty-users.properties
      ```

See the Jetty Authentication documentation for more details.

- **Tomcat**
  - Create or edit your `$CATALINA_HOME/conf/tomcat-users.xml` file. It has entries of the form

    ```
    <user name="principal" password="password" roles="role1, role2, ..." />
    ```

    where:

    - *name* is the user's login id (the principal)
    - *password* is the user's password
    - *roles* are the servlet roles they are assigned upon login; tomcat allows you to specify any number of roles (or no role at all). Fedora currently supports two roles: **fedoraAdmin**, which is the superuser role, and has rights to do everything; and **fedoraUser**, which is a user role, and must be granted permissions by the Policy Enforcement Point to perform actions.

    Sample **tomcat-users.xml** file that contains three users, two of whom are regular users, and the third of whom (fedoraAdmin) is a Fedora superuser:

    tomcat-users.xml

    ```
    <tomcat-users>
      <role rolename="fedoraUser" />
      <role rolename="fedoraAdmin" />
      <user name="testuser" password="password1" roles="fedoraUser" />
      <user name="adminuser" password="password2" roles="fedoraUser" />
      <user name="fedoraAdmin" password="secret3" roles="fedoraAdmin" />
    </tomcat-users>
    ```

  - Configure your Tomcat login realm.
    Modify your file `$CATALINA_HOME/conf/server.xml` file to configure the login realm with the Fedora 4 webapp context:

Tomcat server.xml Realm

```
<Context>
...
  <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
         resourceName="UserDatabase" />
</Context>
```

See the Tomcat Realms documentation for more details.