Configuring run-as and Default Subjects, and principal-role mapping

Introduction

Starting from version 2.0.1, Geronimo adopts the basic principle that all security flows from Subjects that result from logging in to a security realm. In previous Geronimo releases, security information for run-as and default subjects was constructed entirely outside any security realm. As a result of following the new principle, run-as and default identities can now participate fully in security using such features as named credentials to access such external systems as connectors and web services, and the JACC system is now more fully pluggable.

However, because run-as and default subjects now result from logging in to a security realm, to use such a subject you need to supply the login information for each such subject. This information is encapsulated in a CredentialStore. We supply a simple CredentialStore implementation using XML in your Geronimo plan. Note that this includes plain text passwords for the run-as and default subjects. This might not be a suitable implementation for many environments.

Each application can choose to use a default, global, credential store or specify a specific store, perhaps specific to that application.

Configuring a SimpleCredentialStoreImpl

For each Subject accessible through a credential store, you need to specify an id, the realm to log in to, and credentials, which depend on the security realm requirements but are typically the name and password. The schema is as follows:

Error formatting macro: snippet: java.lang.NullPointerException

At the moment, Geronimo supplies callback handlers for name and password. For other security realm requirements (e.g. certificates), you will have to write a callback handler.

A simple example of credential store configuration would look like this:

Credential Store Example

```
    <gbean name="CredentialStore" class="org.apache.geronimo.security.credentialstore.SimpleCredentialStoreImpl"
>
        <xml-attribute name="credentialStore">
            <credential-store xmlns="http://geronimo.apache.org/xml/ns/credentialstore-1.0">
                <realm name="my-properties-realm">
                    <subject>
                        <id>admin-run-as</id>
                        <credential>
                            <type>org.apache.geronimo.security.credentialstore.NameCallbackHandler</type>
                            <value>system</value>
                        </credential>
                        <credential>
                            <type>org.apache.geronimo.security.credentialstore.PasswordCallbackHandler</type>
                            <value>manager</value>
                        </credential>
                    </subject>
                    <subject>
                        <id>user-run-as</id>
                        <credential>
                            <type>org.apache.geronimo.security.credentialstore.NameCallbackHandler</type>
                            <value>user</value>
                        </credential>
                        <credential>
                            <type>org.apache.geronimo.security.credentialstore.PasswordCallbackHandler</type>
                            <value>user-password</value>
                        </credential>
                    </subject>
                    <subject>
                        <id>default</id>
                        <credential>
                            <type>org.apache.geronimo.security.credentialstore.NameCallbackHandler</type>
                            <value>default</value>
                        </credential>
                        <credential>
                            <type>org.apache.geronimo.security.credentialstore.PasswordCallbackHandler</type>
                            <value>default</value>
                        </credential>
                    </subject>
                </realm>
            </credential-store>
        </xml-attribute>
    </gbean>
```

Again, note that the PasswordCallbackHandler value element contains a plain text password for the user.

Configuring your application to use a particular CredentialStore

Note that this aspect of Geronimo security is completely pluggable and only the default implementation is described here.

Geronimo security for JavaEE applications requires including a <security> element in (one of) the GHeronimo plans for your application. This describes the principal-role mappings to connect the Subjects from your security realm to the roles used in the spec deployment descriptors (and annotations). It also describes how to interpret run-as roles as subjects through specifying a credential store and the id and realm for each role used as a run-as. Similarly a default subject can be specified in the credential store.

The schema for security configuration is as follows:

Error formatting macro: snippet: java.lang.NullPointerException

The credential store to use is specified in the credential-store-ref. Normally you only need only supply the name component of the credential store name: for most purposes you are likely to include an app specific credential store in the application plan, but otherwise you need to assure that the credential store gbean is in the ancestor configurations of the application.

A default subject or each run-as role specifies the information needed to get the subject using a subject-infoType element.

Example Security Configuration

```
      <security use-context-handler="false" xmlns="http://geronimo.apache.org/xml/ns/security-2.0">
        <default-subject>
          <realm>my-properties-realm</realm>
          <id>default</id>
        </default-subject>
        <role-mappings>
          <role role-name="Administrator">
            <principal class="org.apache.geronimo.security.realm.providers.GeronimoUserPrincipal" name="system"
/>
          </role>
          <role role-name="User">
            <run-as-subject>
                <realm>my-properties-realm</realm>
                <id>user-run-as</id>
            </run-as-subject>the loi
            <principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal" name="user"/>
          </role>
        </role-mappings>
      </security>
```

The sample above shows the simplest principal-role mapping: you specify the principal class and name for each principal that maps to a certain role. Normally this will be entirely sufficient to distinguish principals. However, you might have several login modules or security realms that can produce the same principal but with different meanings. In this case you can include the login domain name or realm name to distinguish the principals.

Additional principal specifications

```
<!-- normal, no domain or realm info -->
<principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal" name="user"/>

<!-- login domain name specified -->
<login-domain-principal domain-name="mydomain" class="org.apache.geronimo.security.realm.providers.
GeronimoGroupPrincipal" name="user"/>

<~-- realm name and login domain name specified>
<realm-principal realm-name="my-properties-realm" domain-name="mydomain" class="org.apache.geronimo.security.
realm.providers.GeronimoGroupPrincipal" name="user"/>
```