FeSL Authentication

Configuration

The default location for the Fedora JAAS configuration file is: $FEDORA_HOME/server/config/jaas.conf. This default location can be overridden by specifying an alternative in the Fedora Web Application's web.xml file.

A detailed tutorial on the JAAS configuration file can be found at:
http://java.sun.com/j2se/1.5.0/docs/guide/security/jaas/tutorials/LoginConfigFile.html.

To provide a basic overview, the structure of the JAAS configuration file is

```
application-name {
  module-class01 mode
  option=value
  option=value
  ...
  option=value;

  module-class02
  mode option=value
  option=value
  ...
  option=value;
};
```

The application-name can be any name and is referenced by the application performing the authentication. The configuration file can contain multiple application-names, each with different configurations.

Each application section can contain one or more login modules. Each login module has a flag which specifies how it will behave and can also have an unlimited number of options. Login modules are executed in sequence as listed in the application section. There are four possible flags that can be used which affect the behaviour of the login modules. Each login module configuration is terminated by a semi-colon ';'.

The flags for the LoginModule are as follows:

FLAG	DESCRIPTION
Required	The LoginModule is required to succeed. If it succeeds or fails, authentication still continues to proceed down the LoginModule list.
Requisite	The LoginModule is required to succeed. If it succeeds, authentication continues down the LoginModule list. If it fails, control immediately returns to the application (authentication does not proceed down the LoginModule list).
Sufficient	The LoginModule is not required to succeed. If it does succeed, control immediately returns to the application (authentication does not proceed down the LoginModule list). If it fails, authentication
Optional	The LoginModule is not required to succeed. If it succeeds or fails, authentication still continues to proceed down the LoginModule list.

A. XmlUsersFile Configuration

This is a basic configuration using the XmlUsersFile authentication module. It authenticates users against the fedora-users.xml file that ships with Fedora as the default authentication mechanism.

```
fedora-auth {
  fedora.server.jaas.auth.module.XmlUsersFileModule required;
};
```

B. LDAP Configuration

When using LDAP authentication there are typically three basic configurations that cover most LDAP deployments.

1. DIRECT BIND
   This configuration provides direct binding to an LDAP server for authentication.

```
fedora-auth {
  fedora.server.jaas.auth.module.LdapModule required
  host.url="ldap://directory.example.org"
  auth.type="simple"
  bind.mode="bind"
  bind.filter="uid={0},ou=people,dc=example,dc=org";
};
```

2. BIND-SEARCH-COMPARE
   Some LDAP configurations have a 'binduser' that has access to people objects and their one-way encrypted passwords. This configuration allows the 'binduser' to connect to an LDAP server, search for the user object based on the users entered credentials, extract the users password from the user object and identify the encryption scheme used. Using this encryption scheme, the provided user password is then also encrypted and the results compared. A match results in successful authentication.
   continues down the LoginModule list.
   Optional
   The LoginModule is not required to succeed. If it succeeds or fails, authentication still continues to proceed down the LoginModule list.

```
fedora-auth {
  fedora.server.jaas.auth.module.LdapModule required
  host.url="ldap://directory.example.org"
  auth.type="simple"
  bind.mode="bind-search-compare"
  bind.user="uid=binduser,ou=people,dc=example,dc=org"
  bind.pass="somepassword"
  search.base="ou=people,dc=example,dc=org"
  search.filter="(uid={0})"
  attrs.fetch="cn,sn,mail,displayName,carLicense";
};
```

3. BIND-SEARCH-BIND
   This configuration is almost identical to the bind-search-compare strategy except that instead of finding and comparing the passwords, once a user object is found a bind is executed using that user and the provided password. This configuration is particularly useful for authenticating against Active Directory where user passwords are available from the initial bind and search.

```
fedora-auth {
  fedora.server.jaas.auth.module.LdapModule required
  host.url="ldap://directory.example.org"
  auth.type="simple"
  bind.mode="bind-search-bind"
  bind.user="uid=binduser,ou=people,dc=example,dc=org"
  bind.pass="somepassword"
  search.base="ou=people,dc=example,dc=org"
  search.filter="(uid={0})"
  attrs.fetch="cn,sn,mail,displayName,carLicense";
};
```

C. Cascading Multiple Authentication Mechanisms

Occasionally, it might be useful to authenticate from multiple sources. You might want to authenticate off an LDAP directory, but have a couple of extra users with admin privileges in the fedora-users.xml file. This is in fact recommended as you can then still get access to your repository in the event of a network failure to your LDAP directory. You also might want to authenticate users from multiple LDAP directories, or any other combination. Achieving this with JAAS is trivial.

The example below demonstrates authenticating first off an LDAP server using a direct bind. If this fails, control is passed on to the XmlUsersFile module to attempt to authenticate with the users credentials there. Note the flags for these modules are set to 'sufficient'. This means that only one of these modules needs to successfully authenticate a user for authentication to be successful.

```
fedora-auth {
  fedora.server.jaas.auth.module.LdapModule sufficient
  host.url="ldap://directory.example.org"
  auth.type="simple"
  bind.mode="bind"
  bind.filter="uid={0},ou=people,dc=example,dc=org";

  fedora.server.jaas.auth.module.XmlUsersFileModule sufficient;
};
```

UserServlet

A servlet is provided that produces an XML representation of the authenticated user, along with their attributes. The XML format produced is similar to that of the fedora-users.xml file and is structured as follows:

```
<user id="userid">
  <attribute name="attributename1">
    <value>value1</value>
    <value>value2</value>
  </attribute>
  <attribute name="attributename2">
    <value>value1</value>
  </attribute>
</user>
```

This servlet is configured by default to be accessible via http://server:port/fedora/user and should always be protected by the JAAS authentication filter.

The purpose of this servlet is to provide applications the ability to access user attributes and authenticate users without the need to implement security infrastructure on the application end. For example, attributes such as email and display names can be obtained by applications without them having to configure and use an LDAP based data/authentication store. This also means that the authentication is handled at a single point rather than at the Fedora end and the application end. This will allow easier development of front end applications and remove the duplication of security infrastructure. It also means that the authentication mechanisms are client system agnostic.