

Authentication realms

by SuperBonBon

1. JAFS Realms

JAFS has been designed to use JAAS Pluggable Authentication Modules architecture for good modularity and compatibility with all existing JAAS login modules.

If you need your own solution simply create your own JAAS compatible and plug it into the JAFS config file.

JAFS currently ships a very basic login module (*net.sbbi.jafs.auth.XmlLoginModule*) to authenticate users against informations (user names/passwords) contained into an XML file (default file in config/xml_auth_realm.xml). The passwords are hashed with SHA-256 (or MD5 if no SHA-256 available within the JRE) into the file for security considerations.

You can simply open the file to add or remove users (the grammar is straightforward) or use the JMX MBean to create/remove users within the file.

The XML realm contain two special passwords :

1. \$EMAIL : Accepts any password who looks like an email (foo@bar.com).
2. \$ANY : Accepts any password.

2. How do I set up the whole thing

You'll need to edit the service configuration file [auth-realms](#) entries to define the JAAS login modules you want to use :

```
<auth-realms usable="true">
  <login-configuration name="xml" OTPEnabled="true">
    <login-module class-name="net.sbbi.jafs.auth.XmlLoginModule" flag="Required">
      <setting name="fileLocation">../config/xml_auth_realm.xml</setting>
      <setting name="name">XML realm</setting>
      <setting name="JMXEnabled">true</setting>
      <setting name="JMXServerName">Jafs</setting>
    </login-module>
  </login-configuration>
  <login-configuration name="nt" OTPEnabled="false">
```

```
<login-module class-name="com.tagish.auth.win32.NTSystemLogin" flag="Required"
  <setting name="defaultDomain">default</setting>
  <setting name="returnNames">false</setting>
  <setting name="returnSIDs">true</setting>
</login-module>
</login-configuration>
...
</auth-realms>
```

The authentication realms [usable](#) setting let you define if you want to use the login modules configuration defined in the JAFS service config file or by the regular JAAS login modules configuration file defined by the *java.security.auth.login.config* system setting.

Each [login-configuration](#) let you define a set of JAAS [login modules](#) that the user will need to authenticate against to pass or fail the global authentication process on the server.

Each login module has a [flag](#) to indicate what will be the global authentication process result if the authentication fails with one or more login module of the modules list.

Each login modules set is [named](#) so that it can be used either by a [server](#) or by the admin console [realm setting](#). Finally each login modules have their own [settings](#) to work properly.

3. OS users based authentication

You can authenticate users against Windows and Linux operating system

For Windows based user authentication use the *com.tagish.auth.win32.NTSystemLogin* login module, a config sample is already available in your service config file.

Warning:

The NTSystemLogin does NOT work when the server is started as a windows system service.

For Linux/Unix based user authentication, things are a little bit harder since you'll need to setup a radius server and use the *net.sbbi.jafs.auth.RadiusLoginModule* login module, a config sample is available in your service config file too.

Mac OS users cannot be authenticated for the moment.

4. One Time Password (OTP,S/KEY) support

You can enable or disable OTP support for JAAS login modules set with the [OTPEnabled](#) setting. However please take note that you will need to implement the *net.sbbi.jafs.auth.OTPLoginModule* interface for each login modules of your [login-configuration](#) list if you want such functionality. This restriction is due to an API

Authentication realms

restriction of JAAS that does not allow to retrieve a password for a given user name which is unfortunately needed by OTP.

Currently the XML based (*net.sbbi.jafs.auth.XmlLoginModule*) login module supports this interface. Once a login-configuration is set to be OTP enabled, simply assign its name to the [server authRealm name](#) server configuration setting to enable OTP functionality on the server.

Warning:

When OTP support is enabled the passwords in the XML file for the *net.sbbi.jafs.auth.XmlLoginModule* login module **cannot be encrypted** with SHA-256 (default module behaviour when OTP is disabled, all passwords are encrypted with SHA-256) since OTP needs to retrieve the original user password to work and SHA-256 hash cannot be reversed to their original value (that's the goal, when the XML file is edited the password can't be stolen). The login module will refuse to work as long as it detects SHA-256 encoded passwords into the XML file. You'll need to edit the file and revert the MD5 encoded passwords to their original values to make the OTP functionality work.

5. Radius servers login

You can use the *net.sbbi.jafs.auth.RadiusLoginModule* login module to connect to a remote Radius authentication server. You can take a look at the service configuration file to learn about the configuration parameters required to make it work.