Obscuring Passwords

(Applies to geronimo 2.0.2 and possibly later)

Several geronimo configuration files and login module backing stores can contain passwords, and some users want these to be obscured to various extents. I use the word "obscure" since geronimo needs to read them to function, so anyone with file system access to a geronimo installation will be able to figure out what any keys are to decode them anyway. Unless we require credentials to start geronimo there is no way to eliminate storing keys somewhere as unprotected as the geronimo installation.

There are two kinds of places passwords are stored. One is geronimo configurations such as deployment plans, the var/config/config.xml file, and potentially var/config/config-substitutions.properties (currently not used for this purpose). At the moment only var/config/config.xml is obscured. The other is login module backing stores such as for the properties file and sql login modules. Currently the ldap login module does not support this style of obscurantism. Note that these backing stores can also use password hashing, which I am not discussing here. The geronimo configuration files cannot use password hashing because geronimo needs to use these passwords to obtain access to various protected resources rather than just compare a supplied password with a stored password.

This obscuring method, in short, is applied to gbean attributes named "password" in config.xml and the passwords stored in the properties file and sql login module backing stores.

By default, geronimo uses AES with a hardcoded key. When the server starts (config.xml) or when a login module starts, any unobscured password is replaced by

{Simple}<encrypted password>

This prevents someone from verbatim copying a password out of one of these files, and keeps your passwords available (the key is in svn even if you manage to lose it locally) but of course anyone can look up the key and decode the password.

If you want more obscuring, you run the risk of losing your key and making all your passwords completely unusable.

In general to install a different obscuring strategy you need to write a GBean implementing the org.apache.geronimo.util.Encryption interface (in the geronimo-util module). When it starts it will register with the EncryptionManager and re-encrypt all the existing encrypted passwords and be used for all future password encryption/decryption.

We supply one such gbean, org.apache.geronimo.system.util.ConfiguredEncryption in the geronimo-system module. The gbean configuration needs to include the location of the key and a reference to ServerInfo. The location will be resolved with respect to the server location using ServerInfo. If the key is missing it will be created using SecureRandom seeded with the current time. As with the default, this uses AES.
One easy way to install this gbean is to include the following in the rmi-naming section of var/config/config.xml:

```
<gbean name="ConfiguredEncryption">
    <attribute name="path">var/security/ConfiguredSecretKey.ser</attribute>
    <reference name="ServerInfo"><pattern><name>ServerInfo</name></pattern></reference>
</gbean>
```

You will need to replace the version 2.0.2 with whatever the version of your rmi-naming plugin is.

Once again, note that if you lose the key file, in this case var/security/ConfiguredSecretKey.ser, your passwords will be completely unrecoverable.