Portal Security Upgrade

All sensitive portal configuration data has been removed from the source code and configuration files, and stored in one cleartext file containing the encryption key, named /etc/properties/cso-secretkey.properties, and one encrypted file named /etc/properties/cso-sensitive.properties.

The folder /etc/properties will need to exists on the target server, and the two files mentioned above placed into the folder. The file cso-secretkey.properties will need file permissions that restrict reading to the tomcat user, and the root user only. The properties file with the sensitive value does not have to be tightly controlled. All individuals that should not be able to edit or read the key file, should not be given those privileges.

Encryption is done using the Jasypt encryption library, from [www.jasypt.org](http://www.jasypt.org). This library allows encrypting portions of a standard properties file. An example of the file looks like this:

gwuser=ENC(X+RqjvFCPJFvV57jnwKhSg==)

gwpassword=ENC(2UbkjmkGeTfVp1ROWVd1Qg==)

ldapAdminPass=ENC(G0v0hBGJ1n0u5Wt4JGZy73u4FujhXld9)

okGovLogin=ENC(ToVErR35CIRe/JL5n7z6s2k3o3gKO3t/)

okGovPassword=ENC(8ztJfnUulaKiuaFG+gJs611xGhrIqqo4)

Notice the ENC() section, this is the part that contains the encrypted value. Jasypt has a set of programs called CLI Tools (see <http://www.jasypt.org/cli.html> ). These tools allow encrypting a string using a secret key, and the value to be encrypted. Whenever the encryption key or sensitive values are changed, the encrypt script will be used to change the sensitive values in the properties file.

The gwuser value was encrypted using the cli tools like this:

encrypt input=su key = secretkey

The value that it returns for this is X+RqjvFCPJFvV57jnwKhSg==.

Encryption is performed at the command line with the encrypt.bat/encrypt.sh command.

**Usage**:

$ ./encrypt.sh

Using classpath: .:./lib/icu4j-3.4.4.jar:./lib/jasypt-1.8.jar

USAGE: encrypt.sh [ARGUMENTS]

\* Arguments must apply to format:

"arg1=value1 arg2=value2 arg3=value3 ..."

\* Required arguments:

input

password

\* Optional arguments:

verbose

algorithm

keyObtentionIterations

saltGeneratorClassName

providerName

providerClassName

stringOutputType

**Examples**:

$ ./encrypt.sh input="This is my message to be encrypted" password=MYPAS\_WORD

----ENVIRONMENT-----------------

Runtime: Sun Microsystems Inc. Java HotSpot(TM) Client VM 1.6.0\_03-b05

----ARGUMENTS-------------------

input: This is my message to be encrypted

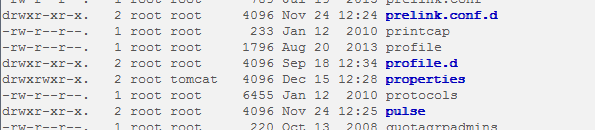
password: MYPAS\_WORD

----OUTPUT----------------------

k1AwOd5XuW4VfPQtEXEdVlMnaNn19hivMbn1G4JQgq/jArjtKqryXksYX4Hl6A0e

# File Permissions

the /etc/proprties folder:



The files within the /etc/properties folder:

