Encryption in JasperReports Server 7.5

JasperReports Server 7.5 introduces a change in the use of encryption to improve the management of secured values in the JasperReports Server repository, web application, APIs and import/export processes.

This document is based on:
https://community.jaspersoft.com/wiki/encryption-jasperreports-server-75 which will be updated as more of your use cases become understood.

TL;DR

Here are the 7.5 encryption changes and what you MUST do to have successful deployment and use of the Server. More advanced use cases are covered later in the document.

Keystore files

There are keystore files created during installation and upgrade that now need to be managed carefully. The JasperReports Server will not start and the command line tools will not work if the keystore files are missing or invalid for a given repository.

What you have to do

- Backup and restore your keystore files at the same time as your repository
- Make sure the keystore files are accessible by the web app and command line

Export/Import

Repository exports from one JasperReports Server deployment/repository contain encrypted values that will not automatically import into a different deployment/repository.

- If the keystore files related to an export are not acceptable for the deployment being imported into, the import will fail
- Exports from versions of JasperReports Server 7.2 and below will import into 7.5+ deployments

What you have to do

1. If you want to have export and import work easily across all the JasperReports Server 7.5+ deployments you control:

    - Save the set of keystore files from the first environment you install
    - Configure the keystore files to be used before the install/upgrade process is executed in a new environment. This may require Updating the Keystore files
    - This will work across export/import in the JasperReports Server user interface, REST API and command line
    - Note that, using this approach, secured values from one environment, ie. production, can be successfully imported into other environments. This exposes secured values in a way

that may not be acceptable according to your security requirements.

2. <u>If you want to have an export that is import-able by anyone, including outside your organization:</u>

    - You must use the JasperReports Server command line tools
    - Export from command line:
        - *js-export --<what to export> --output-zip export_what_to_export.zip --keyalias deprecatedImportExportEncSecret*
    - Import from command line:
        - *./js-import.sh --input-zip export_what_to_export.zip --keyalias deprecatedImportExportEncSecret*

Keystore files

The encryption process is based on “keystore related files” on the operating system where the JasperReports Server processes are running:

- .jrsks - Java keystore file
- .jrsksp - keystore properties
    - Java properties file
    - Base64 encoded
    - Managed through provided tools - cannot edit directly
    - Decode to see with `cat ~/.jrsksp | openssl base64 -d`

By default, these are created in the user home directory of the operating system user running install/upgrade.

These files MUST be available to the JasperReports Server and related command line tools in order to encrypt and decrypt values.

The key related files created or present at the installation or upgrade time for a given repository database MUST be used going forward.

- Without those keystore files, secure values in a repository cannot be accessed.
- The keystore files need to be backed up along with the repository.
- A JasperReports Server instance must be given the keystore files for the repository instance it is connecting to.

Keystore files for the JasperReports Server web application

Keystore file locations are defined - in precedence order:

1. “ks” and “ksp” environment variables
2. WEB-INF/classes/keystore.init.properties: ks, ksp properties

This properties file and the keystore files are deployed automatically to the JRS web application via the command line tools.

If you are not using the command line tools to deploy and update the JasperReports Server web application, you will need to manage the deployment of these files and environment variables yourself.

By default during installation and upgrade, the keystore files go into the user home directory. This can cause problems if the web app runs as a different user that the user used to install, ie.:

- Installed/upgraded JasperReports Server as root user
- By default, keystore files were generated into /root
- .jrsksp property ksPath = /root/.jrsks
- The web app is run as tomcat
- Tomcat fails to start, as the tomcat user does not have access to /root.

If the JasperReports Server cannot find the keystore files - maybe because of permissions as noted above, you will get an exception on server start like:

Failed to instantiate [com.jaspersoft.jasperserver.crypto.KeystoreManager]: Please make sure that `create-keystore` was executed; nested exception is java.lang.RuntimeException: KeystoreManager was never initialized or there are errors while instantiating the instance.

To fix this, you need to move the keystore files into a directory that is accessible by the user running the web app process. See Updating keystore files below.

If the keystore files are accessible, but there is a problem with the keystore file location in .jrsksp, you will see exceptions in command line output and JRS log file:

java.lang.Exception: Keystore may have been tempered with.

Keystore files for the command line tools

The keystore files are created and maintained through the command line tools.

The directories where the key related files are stored for the command line tools is defined in the following locations, inspected in order:

1. “ks” and “ksp” environment variables
2. buildomatic/keystore.init.properties: ks, ksp properties
    a. Created when first command is run, based on environment variables or default_master.properties
    b. Delete this if you want to update the location in the environment or default_master.properties or revert to default
3. buildomatic/default_master.properties: ks, ksp properties
4. Coded default: $USER_HOME as per the operating system

ks, ksp variables need to be encoded for the operating system:

- Windows: ks=C\:\\Users\\swood
- Linux: ks=/home/swood
- Mac: ?

Updating the Keystore files

The .jrsksp file includes a full path to the .jrsks file. The encryption logic in 7.5 requires copied .jrsks files need to be in exactly the same directory path as when the related key files were created, and they are accessible by the user running the JRS command line or web app. Ie.

If you want to deploy the keystore files into a given directory ie. /path/to/files

1. Base64 decode the .jrsksp file
2. Update the ksPath property in the decoded file to the full path where you want to deploy ie. ksPath = /path/to/files/.jrsks
3. Base64 encode the updated .jrsksp file

4. Put the updated .jrksp and .jrsks in /path/to/files
5. Update the keystore.init.properties to point to the /path/to/files/.jrsks and .jrsks