Configuring the MobileFirst Server Keystore

Overview

A keystore is a repository of security keys and certificates that is used to verify and authenticate the validity of parties involved in a network transaction. The MobileFirst Server keystore defines the identity of MobileFirst Server instances, and is used to digitally sign OAuth tokens and Direct Update packages. In addition, when a MobileFirst adapter communicates with a back-end server using mutual HTTPS (SSL) authentication, the keystore is used to validate the SSL-client identity of the MobileFirst Server instance.

For production-level security, during the move from development to production the administrator must configure MobileFirst Server to use a user-defined keystore. The default MobileFirst Server keystore is intended to be used only during development.

Notes

- To use the keystore to verify the authenticity of a Direct Update package, statically bind the application with the public key of the MobileFirst Server identity that is defined in the keystore. See Implementing secure Direct Update on the client side (../../application-development/direct-update).
- Reconfiguring the MobileFirst Server keystore after production should be considered carefully. Changing the configuration has the following potential effects:
    - The client might need to acquire a new OAuth token in place of a token signed with the previous keystore. In most cases, this process is transparent to the application.
    - If the client application is bound to a public key that does not match the MobileFirst Server identity in the new keystore configuration, Direct Update fails. To continue getting updates, bind the application with the new public key, and republish the application. Alternatively, change the keystore configuration again to match the public key to which the application is bound. See Implementing secure Direct Update on the client side (../../application-development/direct-update).
    - For mutual SSL authentication, if the SSL-client identity alias and password that are configured in the adapter are not found in the new keystore, or do not match the SSL certifications, SSL authentication fails. See the adapter configuration information in Step 2 of the following procedure.

Setup

1. Create a Java keystore (JKS) or PKCS 12 keystore file with an alias that contains a key pair that defines the identity of your MobileFirst Server. If you already have an appropriate keystore file, skip to the next step.

    > **Note:** The type of the alias key-pair algorithm must be RSA. The following instructions explain how to set the algorithm type to RSA when using the **keytool** utility.

    You can use a third-party tool to create the keystore file. For example, you can generate a JKS keystore file by running the Java **keytool** utility with the following command (where `<keystore name>` is the name of your keystore and `<alias name>` is your selected alias):

```
keytool -keystore <keystore name> -genkey -alias <alias name> -keylag RSA
```

The following sample command generates a **my_company.keystore** JKS file with a `my_alias` alias:

```
keytool -keystore my_company.keystore -genkey -alias my_alias -keyalg RSA
```

The utility prompts you to provide different input parameters, including the passwords for your keystore file and alias.

> **Note:** You must set the `-keyalg RSA` option to set the type of the generated key algorithm to RSA instead of the default DSA.

To use the keystore for mutual SSL authentication between a MobileFirst adapter and a back-end server, also add a MobileFirst SSL-client identity alias to the keystore. You can do this by using the same method that you used to create the keystore file with the MobileFirst Server identity alias, but provide instead the alias and password for the SSL-client identity.

2. Configure MobileFirst Server to use your keystore: in the IBM MobileFirst Foundation Operations Console navigation sidebar, select **Runtime Settings**, and then select the **Keystore** tab. Follow the instructions on this tab to configure your user-defined MobileFirst Server keystore. The steps include uploading your keystore file, indicating its type, and providing your keystore password, the name of your MobileFirst Server identity alias, and the alias password.

When configured successfully, the Status changes to "User Defined". Otherwise, an error is displayed and the status remains "Default".

The SSL-client identity alias (if used) and its password are configured in the descriptor file of the relevant adapter, within the `<sslCertificateAlias>` and `<sslCertificatePassword>` subelements of the `<connectionPolicy>` element. See HTTP adapter connectionPolicy element (../../adapters/javascript-adapters/js-http-adapter/#the-xml-file).