Application Authenticity Protection in Native Android applications

fork and edit tutorial (https://github.ibm.com/MFPSamples/DevCenter/tree/master/tutorials/en/foundation/6.3/authentication-security/application-authenticity-protection/application-authenticity-protection-native-android.html) | report issue (https://github.ibm.com/MFPSamples/DevCenter/issues/new)

This is a continuation of the Application Authenticity Protection (../) tutorial.

Adding required files

From the MobileFirst project's Native API folder, copy the following folders to your native's project lib folder: armabi, armabi-v7a, mips, x86.

application-descriptor.xml

Adding the security test

Modify the application-descriptor.xml file of your application.

Add the securityTest attribute to the Android or iPhone/iPad environment element. For example:

<iphone bundleId="com.worklight.MyBankApp" applicationId="MyBankApp" securityTest="customTests"
version="1.0">

Adding the public signing key

Extract the public signing key of the certificate that is used to sign application bundle (.apk file).

- If building the application for distribution (production), extract the public key from the certificate that is used to sign the production ready application.
- If building an application in the development environment, the default public key that is supplied by the Android SDK can be used. The development certificate can be found in a keystore that is in a {user-home}/.android/debug.keystore file.

The public signing key can be extracted either manually or by using the wizard that MobileFirst Studio provides.

To use the wizard:

- Right-click the Android NativeAPI folder and select Extract public signing key.
- Specify the location and the password of a keystore file, and click **Load Keystore**. The default password for debug.keystore is "android".
- Set the Key alias and click Next.
- A dialog opens that displays the public key.
- Click **Finish** to automatically paste the public signing key to the relevant section of the application-descriptor.xml file.



Add the Application package name by using the Application Descriptor Editor (design view):



Take the Application package name value from the package attribute of the *manifest* node in the **AndroidManifest.xml**.

If you decide to change the value to another, verify that you change it in both locations.

The application-descriptor.xml file can also be edited directly to add the *packageName*: