Application Authenticity Protection

Overview

By issuing an HTTP request, any entity can access the HTTP services (APIs) that IBM MobileFirst Platform Foundation Server offers.
The out-of-the-box Application Authenticity Protection Security Check (../authentication-concepts/) ensures that an application that tries to connect to a MobileFirst Server instance is the authentic one and was not tampered with or modified by a third-party attacker.

To enable Application Authenticity Protection you can either follow the on-screen instructions in the MobileFirst Operations Console → [your-application] → Authenticity, or review the information below.

Availability

Application Authenticity Protection is available in all supported platforms (iOS, Android, Windows 8 Universal, Windows 10 UWP) in both Cordova and Native applications.

> **Note:** Application Authenticity Protection is **not available** in the MobileFirst Development Server. To test, use a remote application server such as a QA, UAT or Production server.

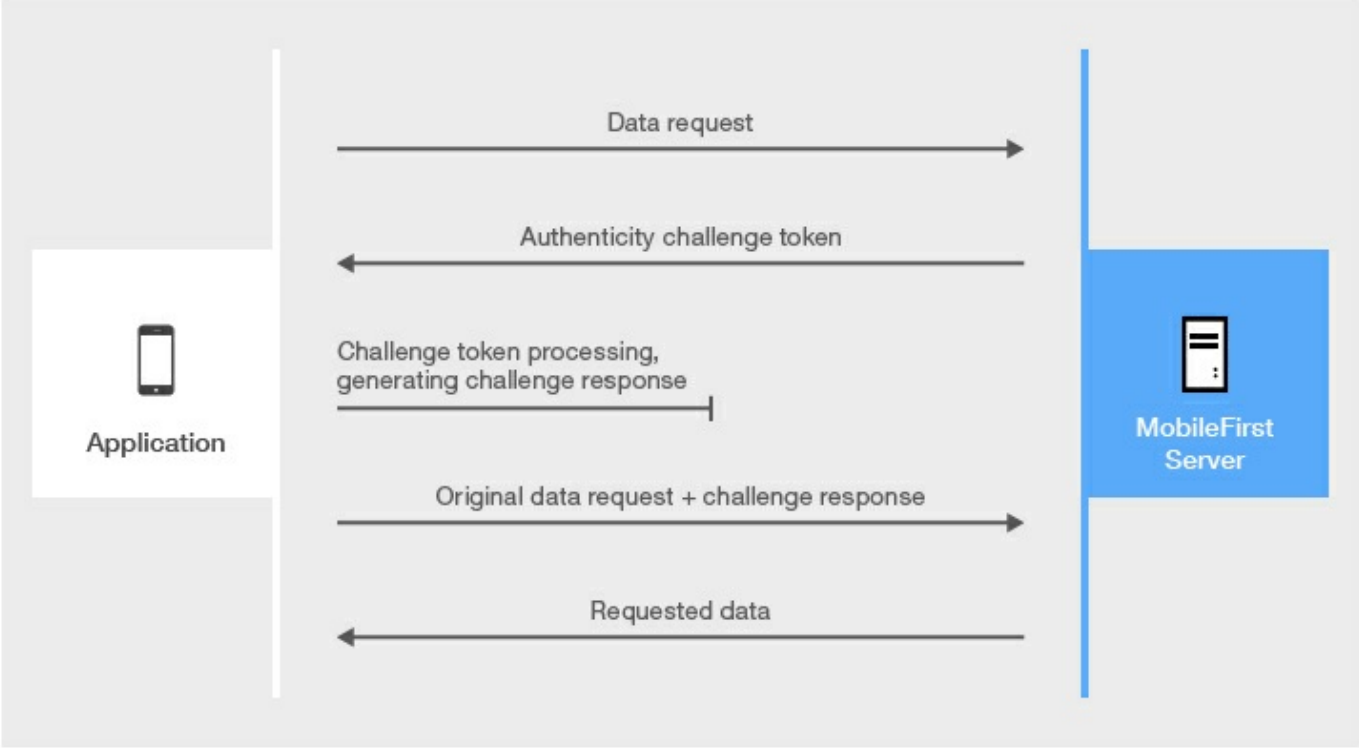
Jump to:

Authenticity Flow

Application Authenticity Protection is based on certificate keys that are used to sign the application bundles. Only the developer or the enterprise who have the original private key that was used to create the application are able to modify, repackage, and re-sign the bundle.

Once an application has successfuly registered with the MobileFirst Server, and passed the Authenticity challenge, an Authenticity token is granted. For as long as the token is valid, the Authenticity challenge will not occur again. See Configuring authenticity (configuring-authenticity) to learn how this can be customized.

TODO: Update with a new diagram from the design team



> The challenge token in the diagram is processed by compiled native code, so that third-party attackers cannot see the logic of this processing.

Enabling Application Authenticity Protection

In order to enable Application Authenticity Protection in your Cordova or Native application, the application's binary file needs to be signed using the MobileFirst-supplied command line tool. Eligible binary files are: `ipa` for iOS, `apk` for Android and `appx` for Windows 8 Universal & Windows 10 UWP.

1. Open **Terminal** and run the command: `java -jar path-to-mfp-server-authenticity-tool.jar path-to-binary-file`

   For example:

   ```
   java -jar /Users/idanadar/Desktop/mfp-server-authenticity-tool.jar /Users/idanadar/Desktop/MyBankApp.ipa
   ```

   The result of the command above is a `.data` file generated next to the `MyBankApp.ipa` file, called `MyBankApp.appAuthenticity.data`.

2. Open the MobileFirst Operations Console in your browser of choice.
3. Select your application from the left-side pane and click on the Authenticiy menu item.
4. Click on "Upload File" to upload the `.data` file.

After uploading the `.data` file Application Authenticity Protection will be enabled for the application.

TODO: add image of where to upload .data file

Disabling Authenticity

In order to disable Application Authenticity Protection, click the "Delete Authenticity File" button.

TODO: add image of where to remove .data file

Configuring Authenticity

The Application Authenticity Protection Security Check has two available properties.
To configure, load the MobileFirst Operations Console and navigate to **[your application] → Security → ???**

- `expirationInSec`: Defaults to 3600 seconds / 1 hour. Defines the duration until the Authenticity token expires.
- `inactivityTimeoutInSec`: Defaults to 0 seconds / no inactivity timeout. Defines the duration of inactivity that if met, will force token expiration.

TODO: add image of where to edit the properties