Securing Cordova Applications

Encrypting the web resources of your Cordova packages

To minimize the risk of someone viewing and modifying your web resources while it is in the .apk or .ipa package, you can use the IBM MobileFirst Foundation mfpdev app webencrypt command or the mfpwebencrypt flag to encrypt the information. This procedure does not provide encryption that is impossible to defeat, but it provides a basic level of obfuscation.

Prerequisites:

- You must have the Cordova development tools installed. This example uses the Apache Cordova CLI. If you use other Cordova development tools, some of your steps will be different. Refer to your Cordova tool documentation for instructions.
- You must have the MobileFirst Platform CLI installed.
- You must have the IBM MobileFirst Foundation plug-in.

The best time to complete this procedure is after finishing your app development and are ready to deploy the app. If you run any of the following commands after you complete the web resources encryption procedure, the content that was encrypted becomes decrypted:

- cordova prepare
- cordova build
- cordova run
- cordova emulate
- mfpdev app webupdate
- mfpdev app preview

If you run one of the listed commands after you encrypt the web resources, you must complete this procedure again to encrypt the web resources.

1. Open a terminal window and navigate to the root directory of the Cordova app that you want to encrypt.
2. Prepare the app by entering one of the following commands:
    - cordova prepare
    - mfpdev app webupdate
3. Complete one of the following procedures to encrypt the content:
    - Enter the following command: `mfpdev app webencrypt`. **Tip:** You can view information about the `mfpdev app webencrypt` command by entering `mfpdev help app webencrypt`.
    - You can also encrypt the web resources of your Cordova packages by adding the `mfpwebencrypt` flag to the `cordova compile` or to the `cordova build` command when you build your packages.
        - `cordova compile -- --mfpwebencrypt` | `cordova build -- --mfpwebencrypt`
          The operating system information in the **www** folder is replaced by a **resources.zip** file that contains the encrypted content.
          If your app is for the Android operating system and the **resources.zip** file is larger than 1 MB, the **resources.zip** file is divided into smaller 768 KB .zip files that are named

resources.zip.nnn. The variable nnn is a number from 001 through 999.

4. Test the application with the encrypted resources by using the emulator that is provided with the platform-specific tools. For example, you can use the emulator in Android Studio for Android, or Xcode for iOS.

Note: Do not use the following Cordova commands to test the application after you encrypt it:

- `cordova run`
- `cordova emulate`

These commands refresh the content that was encrypted in the www folder, and saves it again as decrypted content. If you use these commands, remember to complete the procedure again to encrypt it before you publish the app.

Enabling the web resources checksum feature

When it is enabled, the web resources checksum feature compares the original web resources of an app when it is started to a stored baseline that was captured the first time that app was started. This is a good way of identifying any differences in the app that might indicate that the app was modified. This procedure is compatible with the Direct Update feature.

Prerequisites:

- You must have the Cordova development tools installed. This example uses the Apache Cordova CLI. If you use other Cordova development tools, some of your steps will be different. Refer to your Cordova tool documentation for instructions.
- You must have the MobileFirst Platform CLI installed.
- You must have the IBM MobileFirst Foundation plugin.
- You must add the platform to your Cordova project before you can enable the web resources checksum feature for that operating system by entering the `cordova platform add [android|ios|windows]` command.

To enable the web resources checksum feature for a Cordova app, complete the following steps:

1. In a terminal window, navigate to the root directory of your target app.
2. Enter the following command to enable the web resources checksum feature for an operating system environment of your Cordova app:

   ```
   mfpdev app config [android|ios|windows10|windows8|windowsphone8]_security_test_web_resources_checksum true
   ```

   For example:

   ```
   mfpdev app config android_security_test_web_resources_checksum true
   ```

   You can disable the feature by replacing **true** in the command with **false**.

   > **Tip:** You can view information about the `mfpdev app config` command by entering `mfpdev help app config`.

3. Enter the following command to identify the types of files that you want to ignore during the checksum test:

```
mfpdev app config
[android|ios|windows10|windows8|windowsphone8]_security_ignore_file_extensions [ file_extensio
n1,file_extension2 ]
```

Multiple extensions must be separated by a comma with no spaces between them. For example:

```
mfpdev app config android_security_ignore_file_extensions jpg,png,pdf
```

Important: Running this command overwrites the values that are set.

The more files that the web resources checksum scans for its test, the longer it takes for the app to open. You can specify the extension of a file type to skip, which might improve the speed of starting the app.

Your app has the web resources checksum feature enabled.

1. Run the following command to integrate the changes into your app: `cordova prepare`
2. Build your app by entering the following command: `cordova build`
3. Run your app by entering the following command: `cordova run`