PAYFORT

Fraud Native Mobile SDK

Document Version: 1.2

June, 2017

Contents

1 SDK Overview

Fraud SDK is the module used to generate the device fingerprint (information collected about a remote computing device for the purpose of identification). The generated value must be sent in the request map passed to the FORT MOBILE SDK with key equals to 'device_fingerprint'.

The FORT will use the past value to apply the Fraud Red Roles on the transaction requested through the mobile SDK.

2 SDK Infrastructure

Iovation identifies devices through information collected by an iovation Mobile SDK run on an end- user's mobile device. The Mobile SDK inspects the device to generate a blackbox that contains all device information available. This blackbox value represents the device fingerprint requested by the FORT.

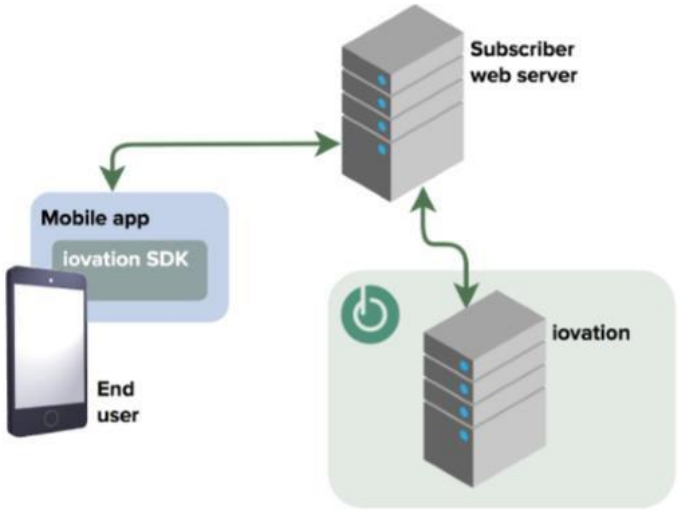


Figure 1: The iovation SDK infrastructure diagram

3 Android

3.1 Android Integration Files and Requirements

SDK Filename	**iovation-android-sdk-2.3.2.zip**
Version	2.3.2
Package	com.iovation.mobile.android.DevicePrint
Android SDK dependencies	Android SDK 2.2 or higher (SDK level 8)
Required Permissions	None
Optional Permissions	• BLUETOOTH • ACCESS_WIFI_STATE • READ_PHONE_STATE • ACCESS_FINE_LOCATION • GET_ACCOUNTS • ACCESS_NETWORK_STATE

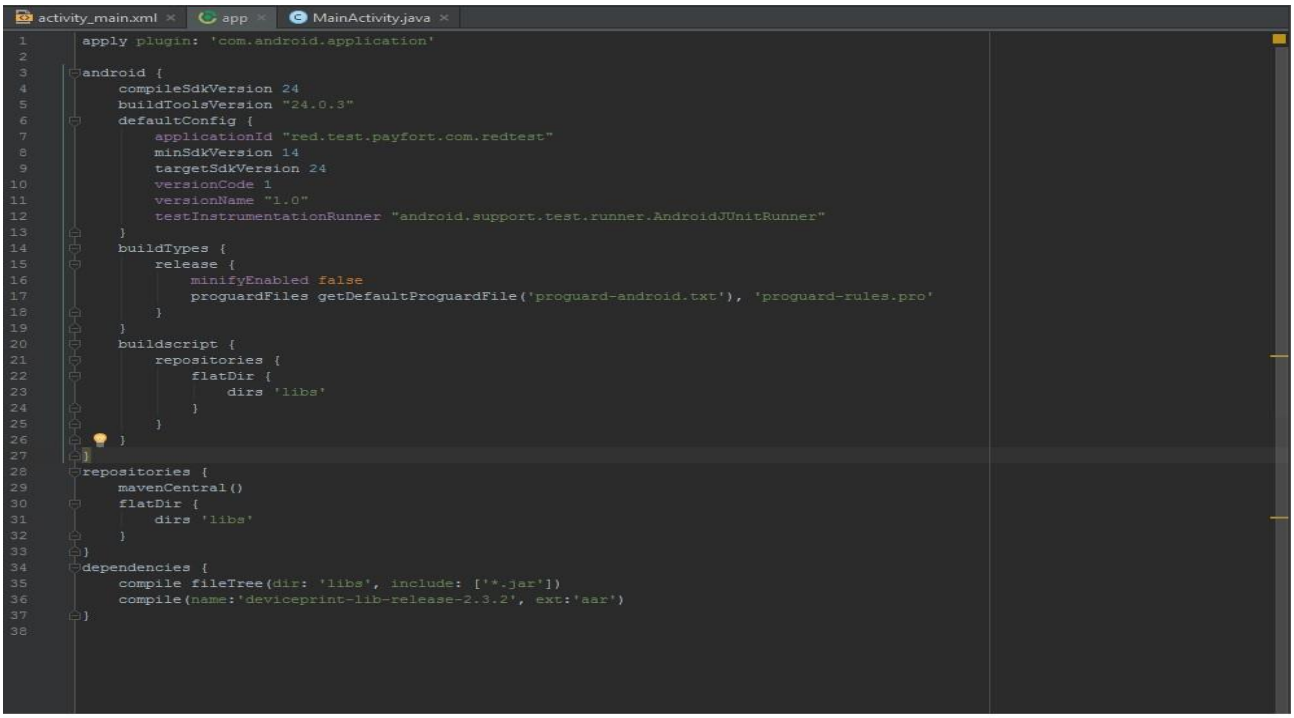
NOTE!

If the permissions listed are not required by the application, the values collected using those permissions will be ignored. The permissions are not required to obtain a usable blackbox, but they do help obtain some unique device information.

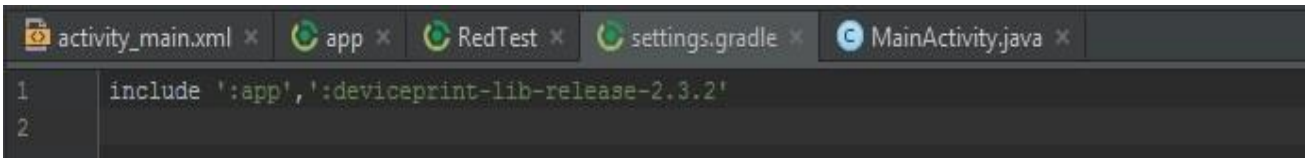
3.2 Installing the SDK for Android

Follow these steps to install Fraud SDK for Android:

1. Add the **"iovation-android-sdk-2.3.2.zip"** file to the **"libs"** directory.
2. After adding the dependencies to the **"gradle.build"** file, it should look like the image below:



3. Add a reference to the **"setting.gradle"** file as the image below:



4. The SDK call use:

```
package red.test.payfort.com.redtest;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import com.iovation.mobile.android.DevicePrint;
import static com.iovation.mobile.android.DevicePrint.getBlackbox;

public class MainActivity extends Activity {

    private TextView value;
    private Button mBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initActivity();
        DevicePrint.start(getApplicationContext());


        mBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String blackbox = getBlackbox(getApplicationContext());
                value.setText(blackbox);
            }
        });

    }

    private void initActivity(){
        this.value = (TextView) findViewById(R.id.value);
        this.mBtn = (Button) findViewById(R.id.mBtn);
    }


}
```


4 iOS

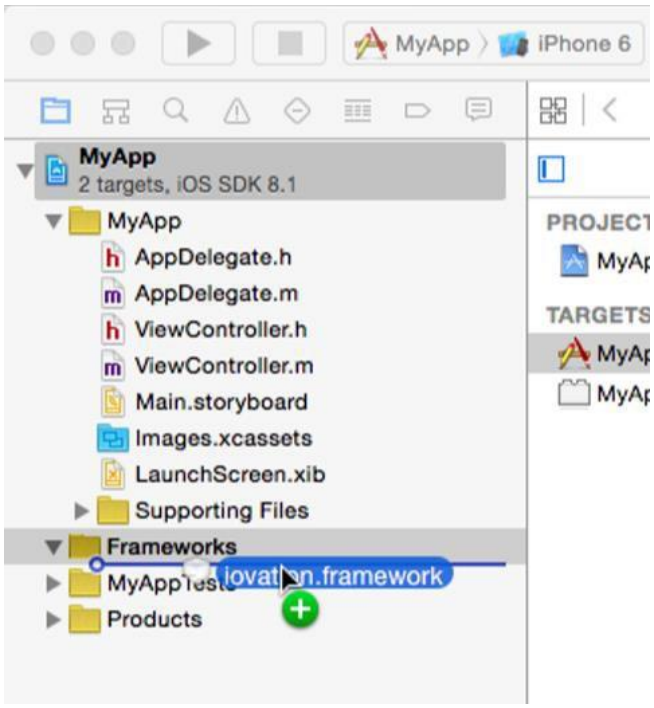
4.1 iOS Integration Files and Requirements

File	**iovation-ios-sdk-4.2.0.zip**
Version	4.2.0
Required OS version	iOS 6.0 and higher
Supported Devices	iPhone 3GS and later, IPod Touch 4th generation and later, iPad 2 and up
Required Frameworks	CoreTelephony, SystemConfiguration, ExternalAccessory
Optional Frameworks	AdSupport, CoreLocation

4.2 Installing the SDK for iOS

Follow these steps to install Fraud SDK for iOS:

1. In the Finder, drag **iovation-ios-sdk-4.2.0.zip** into your project's navigation area in Xcode.



2. In the dialog that appears:
   - Select Copy items if needed to copy the framework file into your project's directory.
   - Check the checkbox for the targets in which you plan to use the framework.
   - Click Finish.

3. Add the following frameworks to your application's target in Xcode:

NOTE!

If the Wireless Accessory Configuration capability in turned on in the Capabilities tab in Xcode 6 or higher, and you don't need it in your app, simply turn it off and add ExternalAccessory.framework again.

   - CoreTelephony.framework
   - SystemConfiguration.framework
   - ExternalAccessory.framework

4. Optionally add these frameworks if your app uses of them:
   - **AdSupport.framework** — if your app displays ads. Do not include if your app does not use the ad framework, because the App Store will reject apps that include the framework but don't use it.
   - **CoreLocation.framework** — if your app uses location monitoring. Do not include this framework unless your application requests geolocation permission from the user.

5. If your app has enabled the Keychain Sharing capability:
   - Add "com.iovation.stm" to its list of Keychain Groups
   - Add the key AppIdentifierPrefix with the string value $(AppIdentifierPrefix) to your app's Info.plist.

NOTE!

How does iovation use iOS Keychain Services? iovation uses iOS Keychain Services to increase device identification persistence. Keychain Services allow keychain items to be stored for each app, or to be shared across multiple apps via keychain access groups. Shared keychain items ensure consistent device recognition across apps with access to a keychain access group. iovation is more likely to persist device recognition across your apps if your app keys, which are specific to your iTunes account, share a common prefix.

For more on keychain access groups, check:
https://developer.apple.com/library/ios/documentation/Security/Reference/keychainser vices/index.htm

4.3 Usage

Use the DevicePrint API to enable the iovation SDK to start collecting blackbox data asynchronously, and to generate a blackbox to submit to your back-end service.

1. import the iovation SDK in your AppDelegate.m file

```
#import <iovation/iovation.h>
```

2. Next, add the following to your app delegate:

```
 -(BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
     [DevicePrint start]
}

  -applicationDidBecomeActive:(UIApplication *)application

{
     [DevicePrint start];
}

  - (void)applicationWillTerminate:(UIApplication *)application

{
    [DevicePrint stop];
}
```

3. When you need to use blackbox just import iovation SDK and call

```
[DevicePrint blackbox];
```

NOTE!

The blackbox returned from +blackbox should never be empty. An empty blackbox indicates that the protection offered by the system may have been compromised.

