Report:

Privacy and Security Assessment of Smart Wearable Gadgets

5 July 2018 (Redacted Version)

Conducted by

CI4S Ltd

info@ci4sec.com

Commissioned by

VPNmentor

www.vpnmentor.com

Privacy and Security Assessment of Smart Wearable Gadgets

Introduction

The global Internet of Things (IoT) market is expected to grow from $157B in 2016 to $457B by 2020, with a compound annual growth rate (CAGR) of 28.5%, according to Forbes 2017 Roundup of IoT Forecasts. The smart wearables market is predicted to take up 3% of that share, growing from 115.4 million shipments in 2017 to 222.3 million by 2021, according to the International Data Corporation (IDC) Worldwide Quarterly Wearable Device Tracker. In recent years the number of wearables that are connected to the internet has significantly grown, making them and their users more susceptible to cyber attacks. By the nature of these devices, they may expose their users to physical harm in addition to the traditional damage posed by these attacks. This project assessed a selection of these consumer products in order to determine if manufacturers are taking the privacy and security of these products as seriously as they should be.

This document details the results of privacy assessments of three smart wearable gadgets: Digitsole Warm series insoles, Modius' headband and Ivy Health's wearable thermometer for kids.

The assessments are intended to evaluate the overall privacy risks the products expose their users to. Each assessed device and application were found to be collecting and exposing personal information, putting their users' privacy at risk.

Methodology

Each assessment consisted of two phases – network traffic inspection and application. In order to facilitate the analysis, the latest play store versions of the applications were downloaded and installed on a real Android 8.0 device, whose WiFi and Bluetooth traffic was then intercepted and scanned for relevant information. To complete the assessments, we examined each product's privacy policy and compared it to the data actually being collected.

Security is rated according to how easily an attacker can achieve control of the wearable device or its companion application and alter their behavior to their needs (the easier it is, the lower the score), while privacy is rated according to the volume and types of data that the application collects about its users (the more data, the lower the core). Security flaws tend to negatively affect privacy, but not necessarily the other way around.

Bluetooth

Having small, smart wearable gadgets as the target of the assessment meant the only interface to the outside world was Bluetooth, and more specifically Bluetooth Low Energy (BLE). BLE allows for authenticated pairing, but none of the gadgets utilized this capability, opening the door for attackers to freely pair with the gadgets and possibly put the users at risk, which in these devices may even cause physical harm.

Firmware Updates

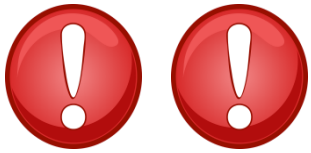
Since none of the gadgets protected their Bluetooth interface from anonymous pairing, they may all be susceptible to an attacker remotely updating their firmware, taking persistent control of the gadget and all data it may contain. Having this in mind, only Digitsole's insoles were found to be capable of having their firmware remotely updated.

Summary

While all assessed devices were found to collect sensitive user data, Modius' application was most restrictive in the information it unintentionally exposed to other applications on the user's device. All three apps were found to utilize either Facebook or Google analytics, which are used to uniquely identify a user across the web. On top of that, they all collect location information and personal user measurements or identifiers, as detailed in the following table:

App	Information Collected	Security Score	Privacy Score
Digitsole Warm	Fine location, age, height, gender, weight, speed, calories burned, steps taken and Facebook analytics	**2/5**	**2/5**
Modius	Coarse location, fingerprint, Facebook analytics and unique mobile device identifiers	**4/5**	**3/5**

Ivy Health Kids	Coarse location, camera, child and parent personal information and temperature measurements, Google analytics and unique mobile device identifiers	**2/5**	**2/5**



1. Digitsole Warm Insoles

Digitsole's smart insoles are Bluetooth-enabled soles which can be put inside shoes in order to enable the user to track their day-to-day and sports activities, and feature the ability to warm up for comfortability.

The assessment was performed on version 1.2 of Digitsole's Android app. The main discovery is that the Digitsole cellphone app exposes personal information, including location and personal details.

Findings

Like many similar products, the insoles do not enforce strict security mechanisms in order to protect themselves. With that being said, the product's privacy policy is clearly phrased to indicate that very little user information is collected and none of it is sold or forwarded to third parties. Furthermore, it details a process to erase any user-collected data saved by Digitsole.

The insoles' mobile app requires a small set of permissions, namely access to location and phone storage, in accordance with the product's privacy policy. Analysis of the app revealed that it collects the following information: location, Facebook profile and friends, movement-related data such as steps taken, calories burned and speed, along with user supplied information such as gender, weight and height. It is worth noting that even though the app's tracking feature can be toggled on and off, if the device's location is turned on and the app is running in the background, it collects location information all the same.

The insoles communicate with the app over Bluetooth. As is common with this type of interface, it implements no authentication, so anyone within Bluetooth connectivity range of the insoles can pair with them and send commands. This can be abused to change the temperature of the insoles, which perform no validation on the temperature set. An attacker can thus set the temperature of the insoles to undesired temperature, though causing damage this way may not be easily feasible due to the amount of time it takes the insoles to warm up.

Furthermore, an attacker that compromises the user's mobile device, can access all of the user's personal data collected by the app, detailed here:

- Data directly given by the user when signing up:
  - Age
  - Height
  - Weight
  - gender

- Data not directly given by the user:
  - Precise location with timestamp
  - Facebook profile and friends
  - Speed
  - Calories burned
  - Steps taken

Digitsole's app collects the latter types of data by using the device's location (GPS) service to get the user's location as well as calculate their average speed over time and steps taken, while also taking into account the collected age, height, weight and gender to calculate the number of calories burned. It also uses Facebook's APIs to get the user's profile and friend data.

Signup data (age, height, weight and gender) is sent to Digitsole's servers once, when registration is complete, or when one of these fields is updated. Real-time data, however, is sent to the servers at a fixed interval every few seconds. All data is sent over an encrypted connection utilizing HTTPS.

Screenshots depicting violations of the user's privacy can be found in appendix A: The first shows the app collecting the user's Facebook data, while the second shows it collecting the user's location whenever the user moves.

In addition to its privacy issues, the Digitsole app also leaves its users susceptible to additional security risks by exposing many unauthenticated services, including a location service and a firmware update service for its insoles. Any application installed on the user's phone can abuse these services to gain

access to the user's location and install rogue firmware on the insoles, without needing any permissions.

Overall, we gave Digitsole's app and device a privacy score of **2 out of 5** and a security score of **2 out of 5.**

2. Modius Headband

This headband is marketed as a weight loss device, intended to change the user's body's weight and appetite by sending electric signals to their brain.

The assessment was performed on version 1.6.0 of Modius' Android app. The main discovery is that the Modius app exposes personal information, including location and personal details, and enables tracking via Facebook integration

Findings

While the permissions requested by Modius' application were more modest than its counterparts, it still required location access along with, surprisingly, fingerprint access (see appendix B.1). It is clearly not easy for an attacker to pair with the headset except with a physical contact to the headset itself.

An attacker who compromises the user's mobile device can thus access the following personal data:

- Coarse location
- Fingerprint
- Facebook tracking
- Weight
- Height
- Waist length
- Body fat percentage
- Modius device usage history
- Personal data, Date of birth, Name, Email

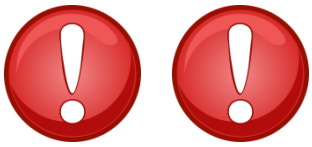
All personal data is sent to Modius' servers once upon registration, while all remaining data is sent whenever the application is used, in regular intervals, as depicted in the screenshots in appendix B. All data is sent over an encrypted channel utilizing HTTPS.

While location access is indeed required for Bluetooth Low Energy scans on Android, this still impacts the user's privacy and so it is reasonable to note this in this report.

Regarding fingerprints, even if this information is not sent to Modius' server, an attacker compromising the user's device will still have access to fingerprint data through the application, since it requires fingerprint access.

With regard to the initialization of the Facebook SDK, as of recent versions, the SDK initializes itself and does not need the main application to initialize it manually.

Overall, we gave Modius' app and device a privacy score of **3 out of 5** and a security score of **4 out of 5.**



3. Ivy Health Kids Thermometer

Ivy Health labs produce smart, portable monitoring devices. Among them is their arm thermometer for babies & small kids. The thermometer connects over Bluetooth to a mobile device app which controls it.

The assessment was performed on version 1.0 of Ivy Health's Kids Android app. The main discovery is that the Ivy Health app exposes personal information, including location and personal details and enables tracking via Facebook analytics.

Findings

The IvyHealth Kids app boasts a wide array of required permissions: read and write access to external storage, camera, location and more. This set of permissions is clearly the largest of all three assessed apps, negatively affecting the user's security and privacy. In much the same way as the two previous apps, Ivy Health's Bluetooth connection did not employ any authentication scheme, therefore allowing any nearby attacker to pair with the thermometer, though effectively posing no danger as the attacker will only be able to read the thermometer's battery level and measured temperature. The app sends all thermal measurements performed by the thermometer to Ivy Health's servers, along with data entered by the user, personally identifying the kids who use the device to take measurements. In addition, the app collects location data along with data uniquely identifying the user's mobile device, for use with Google analytics. To top this off, the app's API and portal (including logon pages) are all served over insecure HTTP, revealing the user's username and password to any eavesdropper.

An attacker who compromises the user's mobile device can thus access the following personal data:

- Coarse location
- All files saved on external storage
- Relationship to kids and other users of the device
- Personally identifying information for all users of the device (both kids and parents):
    - Full name
    - Date of birth
    - Gender
    - Relation to other app users (i.e. daughter, mom,…)
- Temperature measurement history for all users

Personal user data is sent to Ivy Health's servers over insecure HTTP, once when the user registers and whenever any new data is entered or updated, while temperature measurements are sent to the servers every time a measurement occurs.

Technical details can be seen in Appendix C, showing the insecure HTTP portal login page, data sent to Ivy Health's servers upon each measurement and the large set of permissions required by the app to function.

Overall, we gave Ivy Health's app and device a privacy score of **2 out of 5** and a security score of **2 out of 5.**

About

This report was commissioned by VPNmentor.com with all research and analysis conducted by CI4S Ltd.

About VPNmentor.com

VPNmentor offers its users a really honest, committed and helpful tool when navigating VPNs and web privacy.

VPNmentor reviews are not based on advertising; they are based on real experiences, making VPNmentor a truly powerful transparency tool for the internet.

Always go the extra mile. vpnMentor are committed to give their clients with full, detailed advice on anything and everything VPN related.

VPNmentor works diligently to write easy-to-read guides on hard-to-understand subjects. They then translate into 27 languages, so people in Spain, France and Indonesia can equally enjoy the same high quality content as people in the USA.

About CI4S Ltd

CI4S provides cyber-Intelligence and Intelligence-related technologies to the public and private sectors, with decades of experience.

Its customers include large Financial Institutions, Critical Infrastructure, Telecom and Industrial groups worldwide, with a large referral track record.

CI4S relies on best in class in-house experts providing customers with security consultancy services.

Appendix A:

Technical details for the Digitsole Warm insoles assessment

```
75  private void syncUserFriends()
76  {
77    Log.i("FacebookService", "syncUserFriends");
78    Object localObject = ParseUser.getCurrentUser();
79    if ((localObject != null) && (!ParseFacebookUtils.isLinked((ParseUser)localObject)))
80    {
81      Log.e("FacebookService", "User is not linked with Facebook");
82      return;
83    }
84    localObject = new Bundle();
85    ((Bundle)localObject).putString("fields", "name");
86    ArrayList localArrayList = new ArrayList();
87    Log.i("FacebookService", "Requesting user profile");
88    new GraphRequest(AccessToken.getCurrentAccessToken(), "/me/friends", (Bundle)localObject, Htt
89  }
90
91  private void syncUserProfile()
92  {
93    Log.i("FacebookService", "syncUserProfile");
94    ParseUser localParseUser = ParseUser.getCurrentUser();
95    if ((localParseUser != null) && (!ParseFacebookUtils.isLinked(localParseUser)))
96    {
97      Log.e("FacebookService", "User is not linked with Facebook");
98      ParseUser.logOut();
99      notifyDone();
100     return;
101   }
102   Bundle localBundle = new Bundle();
103   localBundle.putString("fields", "name,email,gender,age_range");
104   Log.i("FacebookService", "Requesting user profile");
105   new GraphRequest(AccessToken.getCurrentAccessToken(), "/me", localBundle, HttpMethod.GET, new
106 }
```

Figure A.1 - The Digitsole app collecting Facebook data

```
237   }
238
239   protected void onHandleIntent(Intent paramIntent)
240   {
241     Log.d("LocationService", "onHandleIntent:" + paramIntent);
242     if (paramIntent == null) {}
243     String str;
244     do
245     {
246       return;
247       str = paramIntent.getAction();
248       if ("com.digitsole.action.START_RECORDING".equals(str))
249       {
250         handleStartRecording();
251         return;
252       }
253       if ("com.digitsole.action.STOP_RECORDING".equals(str))
254       {
255         handleStopRecording();
256         return;
257       }
258       if ("com.digitsole.action.LOCATION_UPDATED".equals(str))
259       {
260         handleLocationUpdated(getLocation(paramIntent));
261         return;
262       }
263       if ("com.digitsole.action.STATE_CHANGED".equals(str))
264       {
265         handleStateChanged();
266         return;
267       }
268     } while (!"com.digitsole.action.GET_ALTITUDE".equals(str));
269     handleGetAltitude(paramIntent.getStringExtra("com.digitsole.extra.LOCATION_URI"));
```

Figure A.2 – The Digitsole app collecting location data

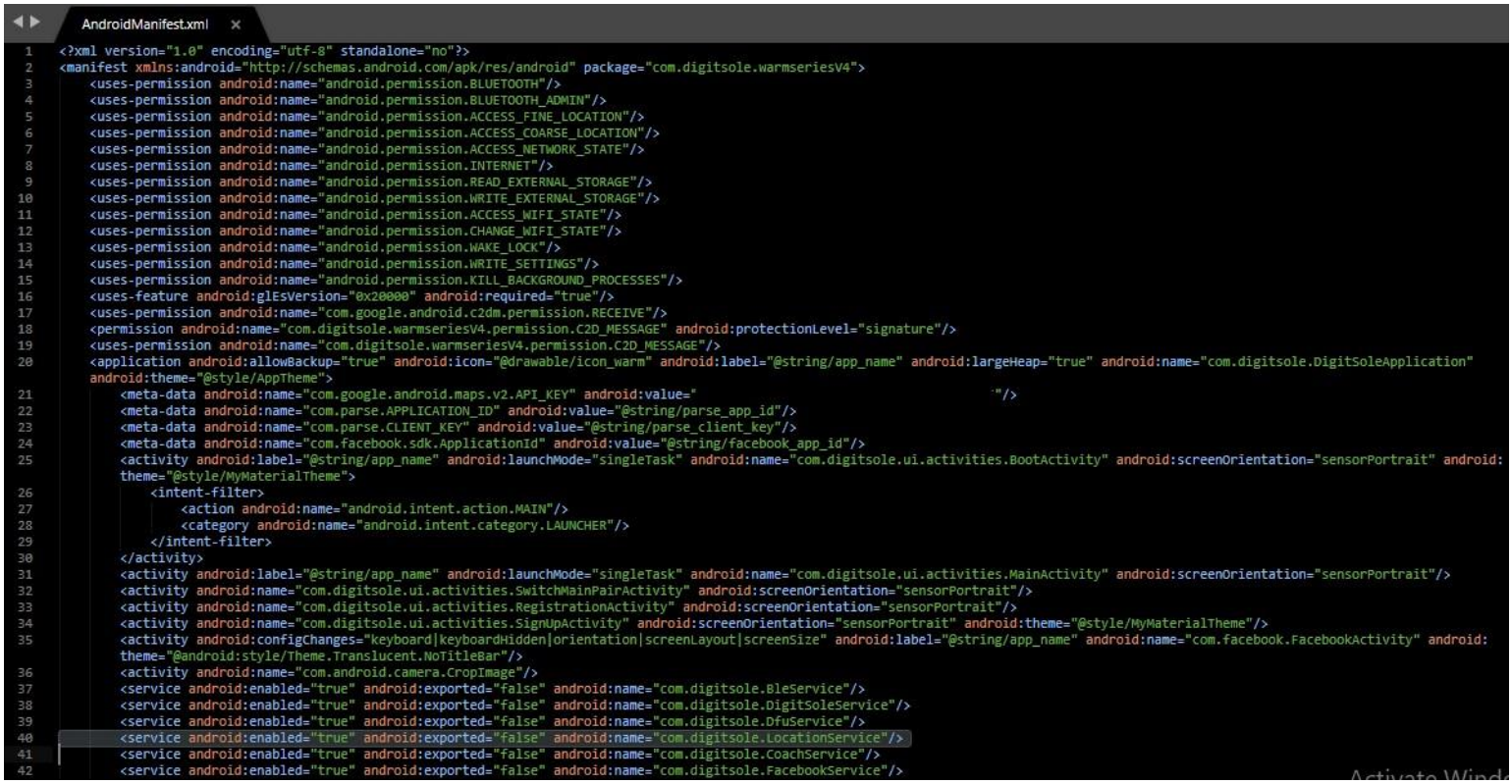


Figure A.3: Large set of permissions required by the Digitsole app, alongside the unauthenticated services it exposes

Appendix B:

Technical details for the assessment of Modius' headband

Figure B.1: Modius' app both integrates with Facebook and requires location access

```
1  package com.neurovalens.modius.models;
2
3  import com.google.gson.annotations.*;
11
12 public class NVSessionLog extends RealmObject implements NVSessionLogRealmProxyInterface
13 {
14     @SerializedName("ble_logs")
15     private RealmList<NVBleLogEntry> bleLogs;
16     private Date end;
17     @GsonExclude
18     @Ignore
19     public boolean isActive;
20     @GsonExclude
21     public boolean isDummySession;
22     @GsonExclude
23     @PrimaryKey
24     private String localID;
25     @SerializedName("pk")
26     private long serverID;
27     private Date start;
28     @SerializedName("user_tz")
29     @Ignore
30     private String userTz;
31
32     public NVSessionLog() {
33         if (this instanceof RealmObjectProxy) {
34             ((RealmObjectProxy)this).realm$injectObjectContext();
35         }
36         this.isActive = false;
37     }
```

Figure B.2: Data that the app sends to Modius' servers, including the headband's session logs

```
1  package com.neurovalens.modius.models;
2
3  import io.realm.*;
10
11 public class NVBodyMeasurement extends RealmObject implements NVBodyMeasurementRealmProxyInterface
12 {
13     @SerializedName("body_fat")
14     private float bodyFat;
15     @GsonExclude
16     public boolean isDummy;
17     @GsonExclude
18     @PrimaryKey
19     private String localID;
20     @SerializedName("pk")
21     private long serverID;
22     @SerializedName("user_tz")
23     @Ignore
24     private String userTz;
25     private float waist;
26     private float weight;
27     private Date when;
28
29     public NVBodyMeasurement() {
30         if (this instanceof RealmObjectProxy) {
31             ((RealmObjectProxy)this).realm$injectObjectContext();
32         }
33         this.realmSet$localID(UUID.randomUUID().toString());
34     }
35
```

Figure B.3: Data that the app sends to Modius' servers, including body fat, waist length, weight and date

Appendix C:

Technical data for the assessment of IvyHealth's thermometer

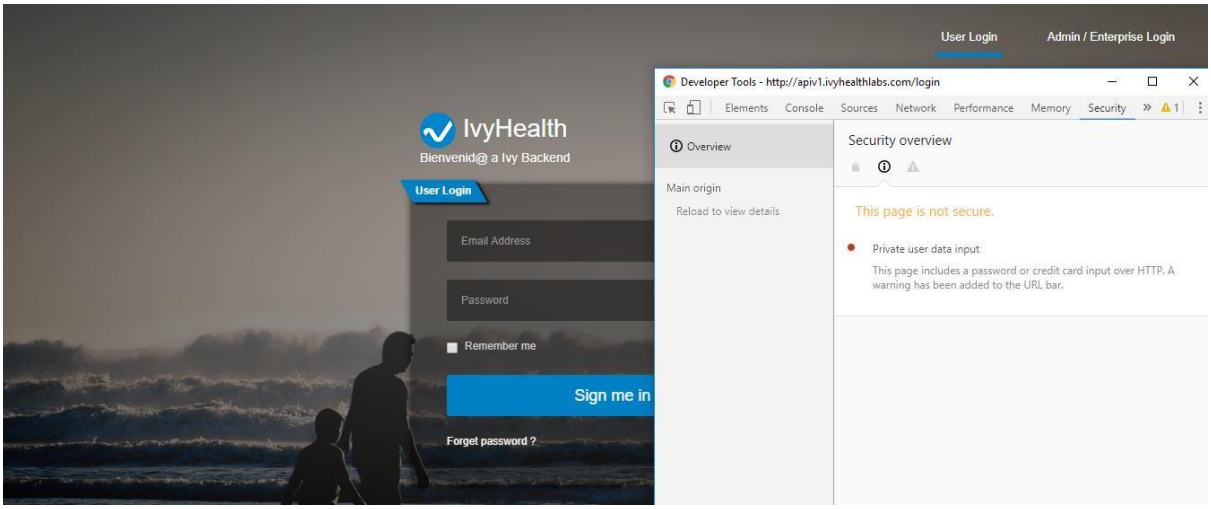


Figure C.1: IvyHealth's app login page is served over insecure HTTP

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.ivy.app">
    <permission android:label="my_permission" android:name="receiver_permission" android:protectionLevel="dangerous"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.BLUETOOTH"/>
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-feature android:name="android.hardware.bluetooth_le" android:required="true"/>
    <uses-feature android:name="android.hardware.camera" android:required="true"/>
    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
    <meta-data android:name="android.support.VERSION" android:value="25.3.0"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <application android:allowBackup="true" android:fullBackupContent="@xml/backup_descriptor" android:icon="@mipmap/ic_launcher" android:label="@string/app_name" android:name="
    ui.activities.IvyApp" android:theme="@style/AppTheme">
        <meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_version"/>
        <meta-data android:name="com.samsung.android.health.permission.write" android:value="com.samsung.health.weight;com.samsung.health.blood_pressure"/>
```

Figure C.2: Vast set of permissions required by IvyHealth's app, including camera and location access

```
package com.ivyhealth.ivyhealthkids.network.model.measurement;

import com.google.gson.annotations.SerializedName;

public class SendMeasurementRequest
{
  @SerializedName("ble_device_id")
  private String bleDeviceId;
  @SerializedName("client_build_number")
  private String clientBuildNumber;
  @SerializedName("client_platform_id")
  private String clientPlatformId;
  @SerializedName("client_platform_version")
  private String clientPlatformVersion;
  @SerializedName("created_at")
  private String createdAt;
  @SerializedName("temperature")
  private float temperature;

  public void setBleDeviceId()
  {
    this.bleDeviceId = "8FE0B908-B214-4256-9EBA-18340F2702D8";
  }
```

Figure C.3: Data that Ivy Health's app collects whenever a measurement takes place

AndroidManifest.xml

```
<activity android:name="ui.measurement.manual.MeasurementManualPressureActivity" android:screenOrientation="portrait" android:theme="@style/AppThemeMeasurementManual"/>
<activity android:name="ui.measurement.MeasurementScaleActivity" android:screenOrientation="portrait"/>
<activity android:name="ui.measurement.MeasurementWristActivity" android:screenOrientation="portrait"/>
<activity android:name="ui.measurement.MeasurementArmActivity" android:screenOrientation="portrait"/>
<activity android:name="ui.measurement.MeasurementGlucoseActivity" android:screenOrientation="portrait"/>
<activity android:name="ui.activities.DisclaimerActivity" android:screenOrientation="portrait"/>
<activity android:name="ui.activities.WebViewActivity" android:screenOrientation="portrait"/>
<activity android:name="ui.charts.ChartBodyActivity" android:screenOrientation="portrait"/>
<activity android:name="ui.charts.ChartPressureActivity" android:screenOrientation="portrait"/>
<provider android:authorities="com.ivy.app" android:exported="false" android:name="database.MeasurementProvider"/>
<meta-data android:name="io.fabric.ApiKey" android:value="a66664289a6c4fabf887fea1bf86332ab3f1c149"/>
<receiver android:enabled="true" android:name="receiver.AlarmReceiver">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"/>
    </intent-filter>
</receiver>
<receiver android:enabled="true" android:name="com.google.android.gms.analytics.AnalyticsReceiver" android:permission="receiver_permission">
    <intent-filter>
        <action android:name="com.google.android.gms.analytics.ANALYTICS_DISPATCH"/>
    </intent-filter>
</receiver>
<service android:enabled="true" android:name="receiver.AlarmService"/>
<service android:enabled="true" android:exported="false" android:name="com.google.android.gms.analytics.AnalyticsService"/>
<receiver android:exported="true" android:name="com.google.android.gms.analytics.CampaignTrackingReceiver" android:permission="receiver_permission">
    <intent-filter>
        <action android:name="com.android.vending.INSTALL_REFERRER"/>
    </intent-filter>
</receiver>
<service android:name="com.google.android.gms.analytics.CampaignTrackingService"/>
<service android:name="network.SyncIntentService"/>
<service android:name="network.Sync_LTC_IntentService"/>
<provider android:authorities="com.ivy.app.crashlyticsinitprovider" android:exported="false" android:initOrder="100" android:name="com.crashlytics.android.CrashlyticsInitProvider"/>
<activity android:label="@string/title_activity_send_feedback" android:name="com.telerik.widget.feedback.SendFeedbackActivity" android:theme="@style/AppCompatTheme"/>
<activity android:label="@string/title_activity_view_feedback" android:name="com.telerik.widget.feedback.ViewFeedbackActivity" android:theme="@style/AppCompatTheme"/>
<activity android:label="@string/title_activity_view_feedback" android:name="com.telerik.widget.feedback.ViewFeedbackItemActivity" android:theme="@style/AppCompatTheme"/>
<activity android:label="@string/title_activity_edit_details" android:name="com.telerik.widget.feedback.EditDetailsActivity" android:theme="@style/AppCompatTheme"/>
<activity android:exported="false" android:name="com.google.android.gms.common.api.GoogleApiActivity" android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
<provider android:authorities="com.ivy.app.firebaseinitprovider" android:exported="false" android:initOrder="100" android:name="com.google.firebase.provider.FirebaseInitProvider"/>
</application>
</manifest>
```

Figure C.4: Ivy Health's app utilizes Google analytics to track users

*** END ***