Security Assessment Report

******* Android client ******* security assessment

****

2014

Authored by: Andrew Rukin

Statement of Confidentiality

This Confidential Information is being provided to ******* as a deliverable of this consulting engagement. The sole purpose of this document is to provide you with the results and recommendations of this engagement. Each recipient agrees that they will follow the distribution restrictions according to the agreement between this consulting agent and *******.

Security Assessment Report

******* Android client ****** Security Assessment

Executive Summary

******* engaged this consultant to conduct a focused Security Assessment Test for the latest publicly available Android client. These build was published on Google Play and identified by version *.*.*.*. The purpose of this engagement was to identify and prioritize the security vulnerabilities on Android client. The engagement was launched on **/**/2014 and included testing, analysis, and documentation.

The following security issues were identified during the course of the Security Assessment:

- Any application may read user's data.
- User's files are saved unprotected on SD card, which make them available to anybody with physical access and in many cases to all applications.
- Users' files are not deleted from SD card on logout, which aggravates previous issue
- Form with search history is available after logout

The following suggestions are recommended to mitigate the findings:

- Limit the accessibility of an app's content provider "com.*******.provider.*******Provider" and to application activity "com.*******.*********"
- Use encryption or application data directory to keep the attachments
- Delete all users' data from SD card on logout

Detailed vulnerability description

Any application may read user' data.

Description

The program does not explicitly assign access permission to a content provider com.*******.provider.*******Provider.

It allows any malicious application, installed on the mobile device silently read user data – assigned tasks, schedules and comments. No permission required, and the ******* application does not even need to be started.

Severity: High

Explanation

Any application can access public components that are not explicitly assigned access permission in their manifest definition. Android content providers are exported by default for applications that set either android:minSdkVersion or android:targetSdkVersion to "16" or lower. For applications that set either of these attributes to "17" or higher, the default is "false".

The following URLs are freely accessible:

- *content://com.*******/accounts*
- *content://com.*******/dashboards*
- *content://com.*******/tasks*
- *content://com.*******/settings*
- *content://com.*******/stream*
- *content://com.*******/stream_entries*

Steps to reproduce

1) Write a program which reads content from above mention content provider. OR
2) Install drozer

   Connect to the device:

   adb forward tcp:31415 tcp:31415

   drozer console connect

Run the following commands to read user data:

*run app.provider.query content://com.*******/accounts*

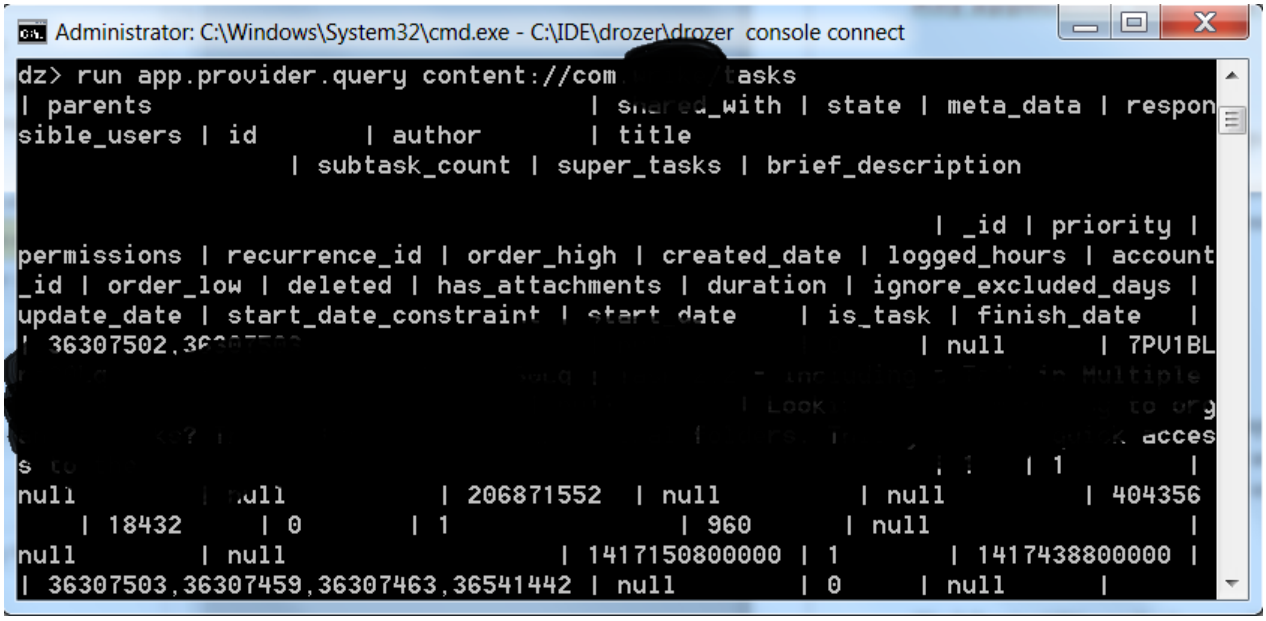
*run app.provider.query content://com.*******/dashboards*

*run app.provider.query content://com.*******/tasks*

*run app.provider.query content://com.*******/settings*

*run app.provider.query content://com.*******/stream*

*run app.provider.query content://com.*******/stream_entries*



Mitigation recommendations

Limit the accessibility of content provider com.*******.provider.*******Provider in AndroidManifest.xml file

*<provider android:label="@string/content_provider" android:name="com.*******.provider.*******Provider" android:exported="**false**" android:authorities="com.*******" android:syncable="true" />*

User's files are saved unprotected on the sd card

Description

If user received a task with attachment – this attachment is saved unprotected in the folder /sdcard/Android/data/com.*******/files/*******/

What “unprotected” means?

1) Anybody with access to the mobile device can read and change it
2) Before Android 4.3 almost ANY program may read and change the attachment

Severity: High

Explanation

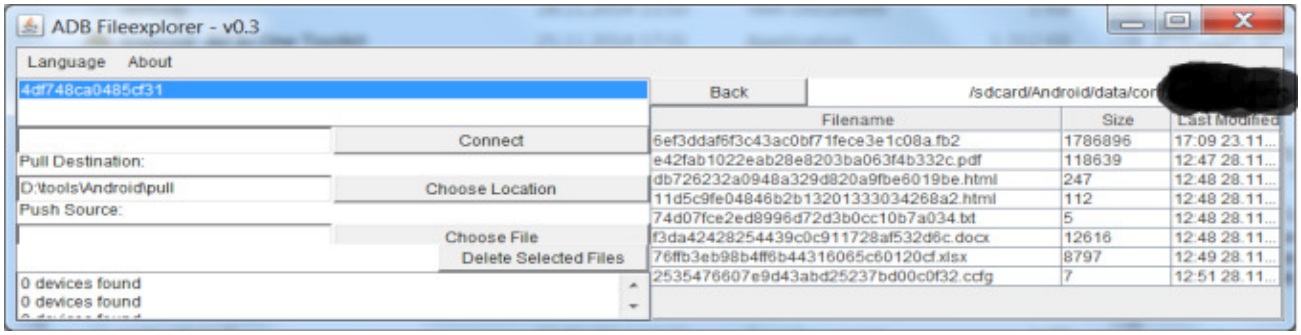
The data on SD card are unprotected, so in case of device loss all the data in the attachments will be available to the person, who obtains the device. Consider what if financial or sensitive information was attached to the task.

1) Anybody with access to the mobile device can read it
2) Before Android 4.3 almost ANY program may read and change the attachment. Some details:

   Files saved to the external storage prior to Android 4.1 are world-readable. Prior to Android 1, files saved to external storage are world-writable. From Android 1 to Android 4.3, only the WRITE_EXTERNAL_STORAGE permission is required for an app to write to any external storage file stored by any app. Starting with Android 4.4, groups and modes of files are created based on a directory structure, which allows an app permission to manage/read/write files within a directory structure based on its package name.

Steps to reproduce

- Using web interface, create a task with attachment, such as .doc, .xls, pdf. Set this task to yourself or other test account.
- Synchronize ******* mobile client.
- You attachments now in /sdcard/Android/data/com.*******/files/*******/ folder, with a temporary name like “f3da42428254439c0c911728af532d6c.doc”



Mitigation recommendations

One might consider encryption when user does not needs the files. It is also possible to use application data directory with permission set to MODE_PRIVATE to keep the files and move them to a publicly available folder when user tries opening it using external program.

User's files are not deleted on logout

Description

If user received a task with attachment – this attachment is saved unprotected in the folder /sdcard/Android/data/com.*******/files/*******/

Consider that user is going to sell the device, so he/she logs out to make sure that confidential data will not be available to new owner. However, the attachments are not deleted from the SD card.

Severity: Average

Explanation

The data on SD card are unprotected, so it is necessary to delete the data on logout.

Steps to reproduce

- Using web interface, create a task with attachment, such as .doc, .xls, pdf. Set this task to yourself or other test account.
- Synchronize ******* mobile client.
- You attachments now in /sdcard/Android/data/com.*******/files/*******/ folder, with temporary name like "f3da42428254439c0c911728af532d6c.doc"
- Logout
- Make sure that your files are still on SD card.

Mitigation recommendations

Delete all users' data from /sdcard/Android/data/com.*******/files/*******/ folder on logout

Form with search history is available after logout

Description

Due to unprotected activity com.*******.SearchActivity it is possible to launch application search form.

Even after logout and without authentication one with a physical access to the device may see user search history.

Severity: Minor

Explanation

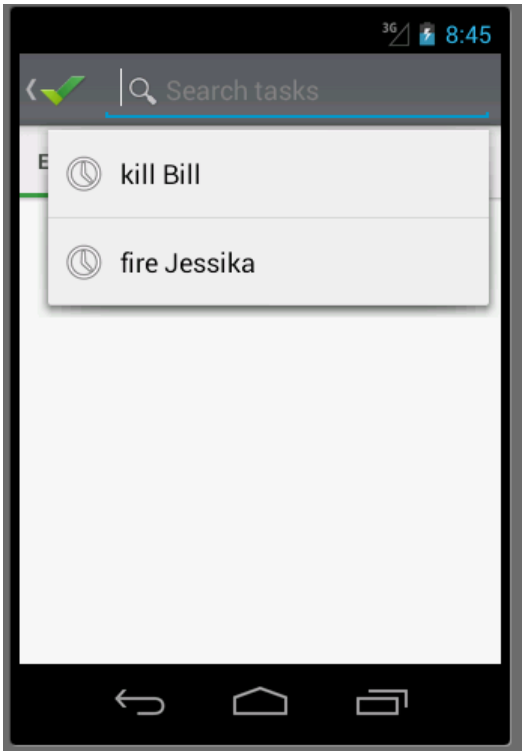
On Android, declaring an intent filter for an activity in the AndroidManifest.xml file means that the activity may be exported to other apps. If the activity is intended solely for the internal use of the app and an intent filter is declared than any other apps can activate the activity for unintended use.

Steps to reproduce

(optional) Logout from application

Use any tool that is able to launch application activity, i.e use drozer: *run app.activity.start --component com.******* com.*******.SearchActivity*

Watch and study search history:



Mitigation recommendations

Restrict access to sensitive activities

In AndroidManifest.xml:

```
<activity android:exported="false" android:label="@string/activity_title_search"
android:name="com.*******.SearchActivity" android:launchMode="singleTop"
android:configChanges="keyboardHidden|orientation">
    <intent-filter>
        <action android:name="android.intent.action.SEARCH" />
    </intent-filter>
    <meta-data android:name="android.app.searchable" android:resource="@xml/task_searchable" />
</activity>
```

Areas Needing Further Review

No code review and White box testing were performed.

Security code review with a static code analyze is highly recommended to locate well-hidden vulnerabilities and ensure protection.