Vulnerability in Scan to PDF for Android

Daoyuan Wu*, Xiapu Luo* and Rocky K. C. Chang
The Hong Kong Polytechnic University
{csdwu, csxluo, csrchang}@comp.polyu.edu.hk
December 14, 2011 PM05:10:28 HKT

Abstract

We found that Scan to PDF (StPDF below) 2.0.4 has a vulnerability that allows a malicious application to access and manipulate user's Google account and scanned files.

1 Application Information

Package Name	com.scan.to.pdf.trial
Full Name	Scan to PDF Free
Version	2.0.4 (the latest version in Android Market)
Category	Productivity
Installs	100,000 - 500,000
Average Rating	4.1/5.0 from 1,857 users

CVE Reference	CVE-2011-4771
Vendor	*Nym Computing*, http://melbina.free.fr/ https://market.android.com/developer?pub=Nym+Computing
Vendor Response	None

2 Description

StPDF exposes the following content provider in the AndroidManifest.xml file, which is not properly protected, as shown in follows:

```
<provider android:name=".providers.DocumentProvider"
android:authorities="com.scan.to.pdf.trial.providers.DocumentsProvider"
/>
```

Thus a malicious application on the same device can access and manipulate user's Google account and scanned files, including images and converted documents through this content provider.

*** authors with equal contributions**

3 Impact

This vulnerability enables an adversary to access and modify user's Google account and scanned images and converted documents, while without being noticed by the user and even without any privilege.

First, StPDF enables the user to upload their scanned images to GDoc (https://docs.google.com) by inputting user's Google account, as shown in Figure 1.

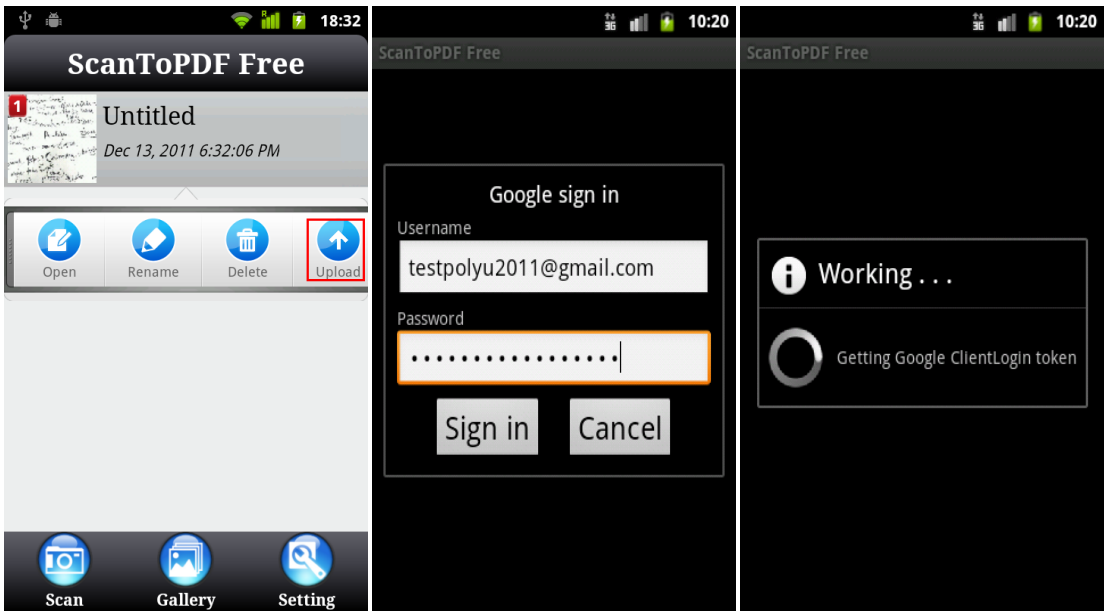

Figure 1: User will input Google account to upload scanned pdf to GDoc.

However, user's Google account above is directly stored in exposed table, "account", as shown in Figure 2. Although StPDF has encrypted password, it's not enough.

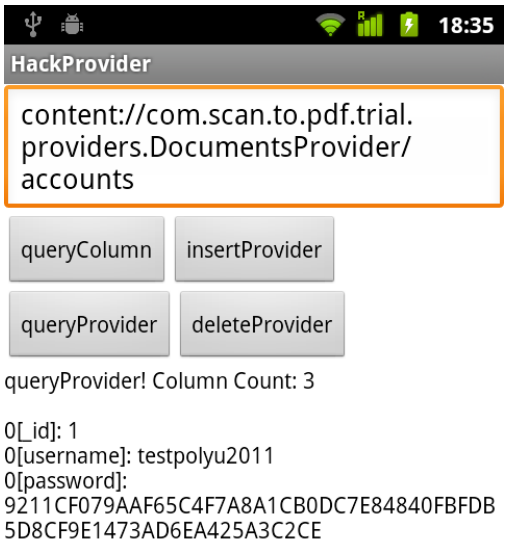

Figure 2: User's Google account is exposed.

On the other hand, all detailed sensitive attributes of scanned images are exposed to public, as

you can see in Figure 3. For example, raw image and thumbnail path in SD Card are not protected well.

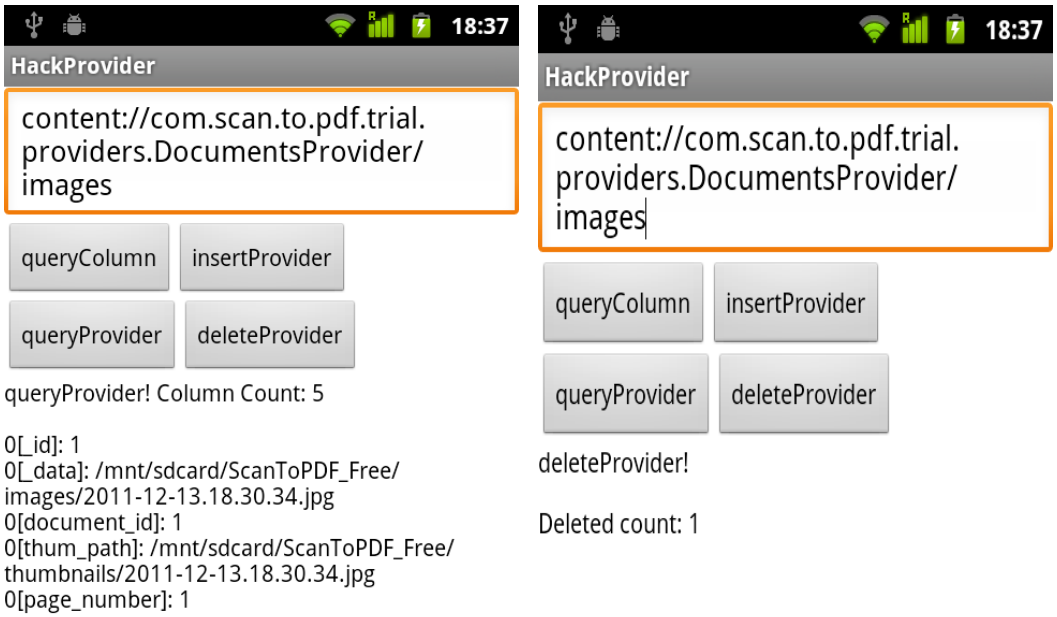

Figure 3: All detailed sensitive attributes of scanned images are exposed to public.

Finally, as shown in Figure 4, all detailed sensitive attributes of converted documents are also exposed, including document title, number of pages, created and modified date, file size and etc.

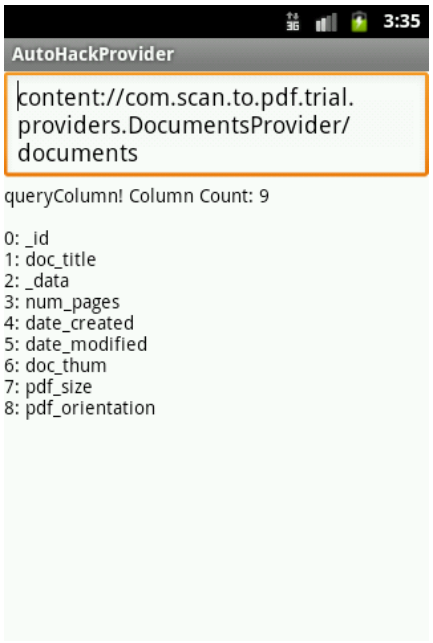

Figure 4: All detailed sensitive attributes of converted documents are also exposed.

4 Solution

We are trying our best to contact *Nym Computing* to fix this security issue. Our advice is to set the permission of these application's content providers properly, or avoid exporting these content providers in the AndroidManifest.xml file. Currently, a user could disable the application temporarily and wait for an official update.

5 Technical Description

Although StPDF only exposes one content provider, "DocumentsProvider", it has 3 exploitable and sensitive tables in total, as shown in the following table:

Content Provider Authority	Table Name
com.scan.to.pdf.trial.providers.DocumentsProvider	accounts
com.scan.to.pdf.trial.providers.DocumentsProvider	documents
com.scan.to.pdf.trial.providers.DocumentsProvider	images

Sample attack codes for "accounts" table in "DocumentsProvider":

```
providerUri =
Uri.parse("content://com.scan.to.pdf.trial.providers.DocumentsProvider/accou
nts")
ContentResolver cr  = this.getContentResolver();

//Insert accounts
ContentValues values = new ContentValues();
....
outUri = cr.insert(providerUri, values);

//Query accounts
Cursor cursor = cr.query(providerUri, null, null, null, null);

//Delete accounts
int nCount = cr.delete(providerUri, null, null);
```