Vulnerability in *Limit My Call* for Android

Daoyuan Wu*, Xiapu Luo* and Rocky K. C. Chang
The Hong Kong Polytechnic University
{csdwu, csxluo, csrchang}@comp.polyu.edu.hk
December 7, 2011

Abstract

We found that *Limit My Call* 2.11 has a vulnerability that allows a malicious application to access and manipulate user's contacts and corresponding calling logs.

1 Application Information

Package Name	com.limited.call.view
Full Name	Limit My Call
Version	2.11 (the latest version in the Android market)
Category	Communication
Installs	50,000 - 100,000
Average Rating	4.2/5.0 from 922 users

CVE Reference	CVE-2011-4703
Vendor	*Nathaniel Kh*, http://nathanielkh.wordpress.com/
Vendor Response	Null

2 Description

Limit My Call exposes the following content provider in the AndroidManifest.xml file, which is not properly protected:

```
<provider
android:name="com.limited.call.model.helper.LimitMyCallProvider"
android:authorities="com.limited.call.view.provider.LimitMyCallProvider"
/>
```

Through the provider com.limited.call.model.helper.LimitMyCallProvider, a malicious application on the same device can query, insert and delete user's contacts and corresponding calling logs.

*** authors with equal contributions**

3 Impact

This vulnerability enables an adversary to access and manipulate user's contacts and corresponding calling logs. For example, as shown in Figure 1, a malicious application on the same device can obtain a contact's name, phone number and photo, without being noticed by the user. Moreover, the vulnerability empowers an attacker to access all calling logs, categorized by callee, including last call time and total call time and times. Thus, a spyware could easily get all detailed contacts and calling logs through this vulnerable provider, while without corresponding android-defined permission.

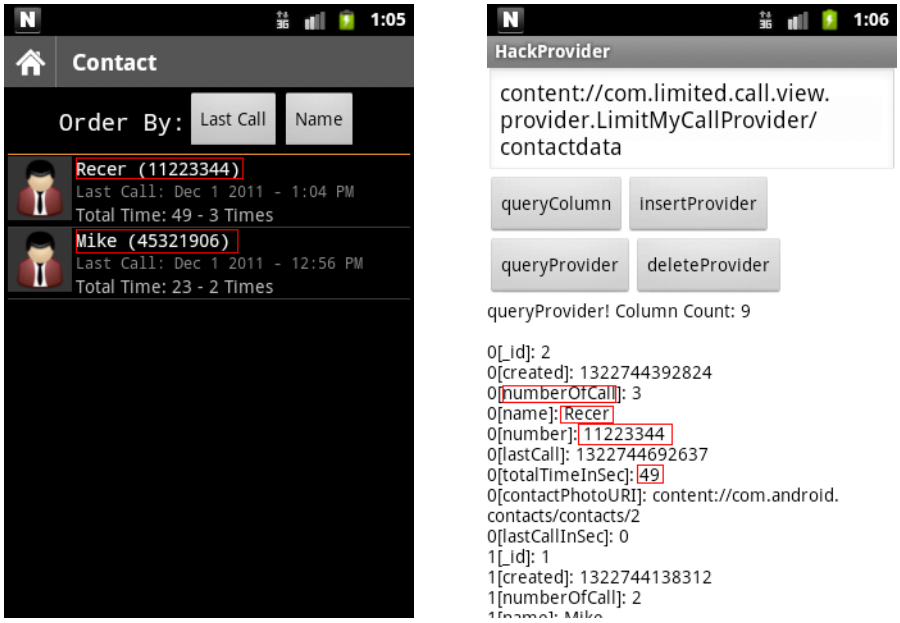

Figure 1: User's contacts and corresponding calling logs are exposed.

4 Solution

We are trying our best to contact *Nathaniel Kh* to fix this security issue. Our advice is to set the permission of this application's content provider 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

The vulnerable content provider and the corresponding table are listed as follows:

Content Provider Authority	Table Name
com.limited.call.view.provider.LimitMyCallProvider	contactdata

Sample attack codes:

```
providerUri = Uri.parse(“content://
com.limited.call.view.provider.LimitMyCallProvider/contactdata”)
ContentResolver cr  = this.getContentResolver();

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

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

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