Vulnerability in QQPimSecure (QQ 手机管家) for Android

Daoyuan Wu*, Xiapu Luo* and Rocky K. C. Chang
The Hong Kong Polytechnic University
{csdwu, csxluo, csrchang}@comp.polyu.edu.hk
December 29, 2011 at 2:51 PM HKT

Abstract

We found that QQPimSecure (QQ 手机管家) has a vulnerability that allows a malicious application to access and manipulate user's sensitive information (e.g., contact list, sms, call log, mms) *protected* by QQPimSecure.

1 Application Information

Package Name	com.tencent.qqpimsecure
Full Name	QQPimSecure ("QQ 手机管家" in Chinese name)
Version	2.3.0, 2.5.1 and 3.1.1
Category	Tools
Installs	100,000 - 500,000
Average Rating	4.6/5.0 from 12,037 users

Pad Version	com.tencent.qqpimsecurepad
Full Name	aPad ("QQ Pad 管家" in Chinese name)
Version	1.1.0
Category	Tools
Installs	500 - 1,000
Average Rating	4.0/5.0 from 2 users

CVE Reference	CVE-2011-4863
Vendor	*Tencent, Inc.*, http://www.qq.com
Vendor Response	

2 Description

QQPimSecure exposes the following content provider in the AndroidManifest.xml file.

*** authors with equal contributions**

```
<provider android:name=".service.QQPimSecureContentProvider"
android:readPermission="com.tencent.permission.CONTENT_PROVIDER"
android:writePermission="com.tencent.permission.CONTENT_PROVIDER"
android:authorities="com.tencent.qqpimsecure.contentprovider" />
```

Although it is protected by "com.tencent.permission.CONTENT_PROVIDER", this permission's protection level is 'Normal' as follows:

```
<permission android:name="com.tencent.permission.CONTENT_PROVIDER"
android:protectionLevel="normal" />
```

A malicious application on the same device can easily obtain this permission because Android automatically gives permission with 'normal' protection level to a requesting application at installation without asking for the explicit approval from the user. Then, the malicious application can access and manipulate user's sensitive information (e.g., contact list, sms, call log, mms) *protected* by qqpimsecure through this content provider.

3 Impact

This vulnerability enables an adversary to access and modify user's sensitive information without being noticed by the user. Moreover, although some sensitive information is protected by password in QQPimSecure's private space as shown in Figure 1, the adversary can still manipulate them without knowing the password as shown in Figure 2 and Figure 3. In other words, the users may assume that their information is secure because of the protection of QQPimSecure, but an adversary can still access the information.

Figure 1: qqpimsecure creates private space to protect the user's information through password.

Figure 2 shows that the contact list and the call log can be obtained by a malicious application by accessing the corresponding tables ‘contactlist’ and ‘pimcalllog’.

Figure 3 shows that the SMS and the protected SMS can be fetched by a malicious application by accessing the corresponding tables ‘’smslog”and “securesmslog”.

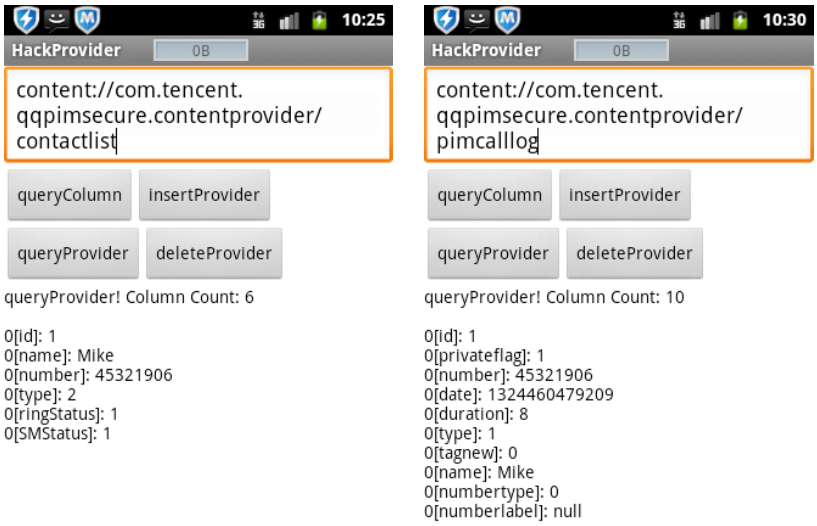

Figure 2: Obtain contact list and call log.

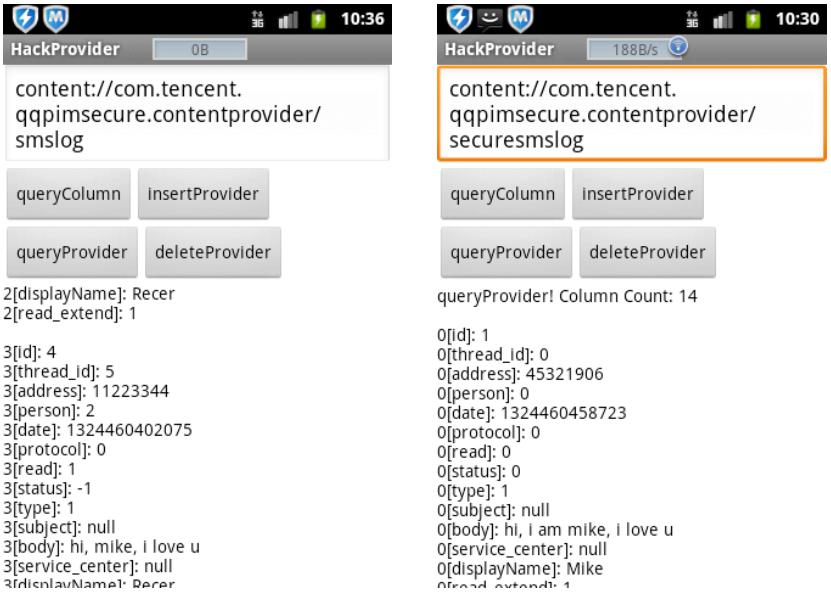

Figure 3: Access SMS and protected SMS.

4 Solution

We are trying our best to contact *Tencent Inc.* to fix this security issue. Our advice is to increase the protection level of this permission or avoid exporting this content provider in the AndroidManifest.xml file. Currently, a user could disable the application temporarily and wait for an official update.

5 Technical Description

Figure 4 shows how a malicious application can obtain the permission without being noticed by the user. More precisely, the user is not alerted when a malicious application requires for such permission and even a user having some knowledge of permission may not know the severe consequence because such permission's protection level is 'normal'.

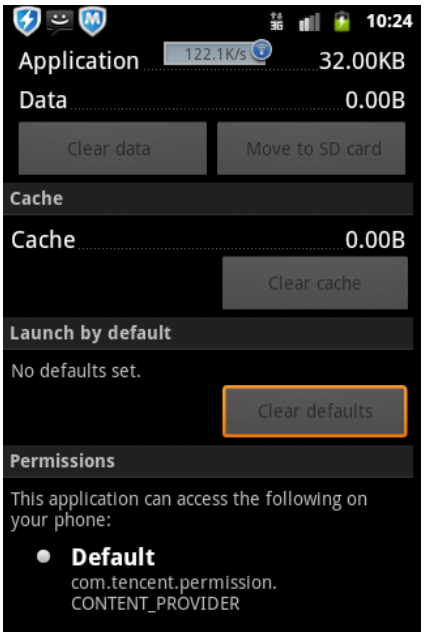

Figure 4: Get the permission of "com.tencent.permission.CONTENT_PROVIDER"

The following table shows the names of tables that can be accessed through qqpimsecure's content provider. These tables store the user's sensitive information.

Content Provider Authority	Table Name
com.tencent.qqpimsecure.contentprovider	keyword
com.tencent.qqpimsecure.contentprovider	contactlist
com.tencent.qqpimsecure.contentprovider	smslog
com.tencent.qqpimsecure.contentprovider	securesmslog
com.tencent.qqpimsecure.contentprovider	pimcalllog
com.tencent.qqpimsecure.contentprovider	smsreport_v2
com.tencent.qqpimsecure.contentprovider	tb_sms_report_temp_v2

com.tencent.qqpimsecure.contentprovider	mms_part
com.tencent.qqpimsecure.contentprovider	secure_mms_part
com.tencent.qqpimsecure.contentprovider	settings

Sample attack codes for manipulating information in the *securesmslog* table:

```
providerUri =
Uri.parse("content://com.tencent.qqpimsecure.contentprovider/securesmslog")
ContentResolver cr  = this.getContentResolver();

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

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

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