Vulnerability in Twidroyd for Android

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

Abstract

We found that Twidroyd 4.3.11 has a vulnerability that allows a malicious application to access and modify user's private twitter information.

1 Application Information

Package Name	com.twidroydlegacy
Full Name	TWIDROYD Legacy
Version	4.3.11 (the latest version in Android Market)
Category	Social
Installs	100,000 - 500,000
Average Rating	4.3/5.0 from 2,610 users

CVE Reference	CVE-2011-4699
Vendor	*UberMedia Inc.*, http://www.ubersocial.com/
Vendor Response	Null

2 Description

Twidroyd exposes the following three content providers in the AndroidManifest.xml file, which are not properly protected, as shown in follows:

- `<provider android:name="com.twidroydlegacy.provider.TweetProvider" android:authorities="twidroid.provider.Tweet" />`
- `<provider android:name="com.twidroydlegacy.provider.DirectMessageProvider" android:authorities="twidroid.provider.DirectMessage" />`
- `<provider android:name="com.twidroydlegacy.provider.SettingsProvider" android:authorities="twidroid.provider.Settings" />`

Thus a malicious application on the same device can access and modify user's private twitter contents through these content providers.

*** authors with equal contributions**

3 Impact

This vulnerability enables an adversary to access and modify user's twitter contents without owning user's twitter account. The exposed private twitter information includes user's all tweets, and even worse, all direct messages are also exposed to public. For example shown in Figure 1, all private attributes of a direct message, such as sender's username, sender's user id, message body, recipient's username and recipient's user id, could be queried from another application on the same device. However, such extremely private information should only be accessible by applications with granted privilege, and meanwhile with user's acknowledgement.

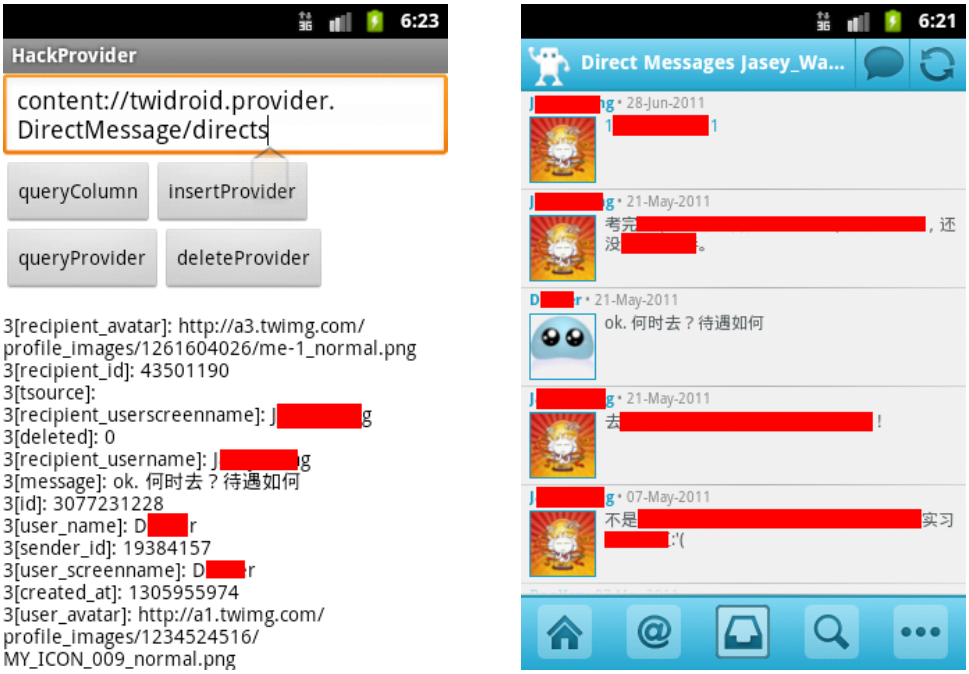

Figure 1: User's all twitter direct messages are exposed.

4 Solution

We are trying our best to contact *UberMedia Inc.* to fix this security issue. Our advice is to set the permission of these application's content providers properly, or just set content providers not exported in the AndroidManifest.xml file. Currently, for a user, just disable the application temporarily and wait for an official update.

5 Technical Description

Among all Twidroyd's content providers, two of them are sensitive, and each has a corresponding table, as shown in the following table:

Content Provider Authority	Table Name
twidroid.provider.DirectMessage	directs

twidroid.provider.Tweet	tweets

When user launches the Twidroyd app at the first time, he or she needs to set up his or her twitter account. After that, Twidroyd would use this account to download user's all latest tweets and direct messages from official twitter web server and store them in the two content providers' tables above. However, as those content providers are not well protected in the AndroidManifest.xml file, thus both "directs" and "tweets" table are exposed and accessible by any other applications without privilege.

As a consequence, a malicious application on the same device could query those "directs" and "tweets" tables without owning user's twitter account, which extremely compromises user's privacy. And even worse, attacker could insert a faked tweet on behalf of user to phish user's friends or just spam in user's social network. Also, attacker could delete user's all tweets and direct messages without user's confirmation.

Figure 2: All private tweets could be queried from another application without privilege.

Sample attack codes:

```
providerUri = Uri.parse("content://
com.twidroydlegacy;twidroid.provider.DirectMessage/directs")
ContentResolver cr  = this.getContentResolver();

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

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

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