# Vulnerability in *CallConfirm* 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 *CallConfirm* 2.0.0 has a vulnerability that allows a malicious application to access and manipulate the allow list and block list set by the user.

## 1 Application Information

| | |
|---|---|
| Package Name | jp.gr.java_conf.ofnhwx.callconfirm |
| Full Name | CallConfirm |
| Version | 2.0.0 (the latest version in the Android market) |
| Category | Communication |
| Installs | 50,000 - 100,000 |
| Average Rating | 4.3/5.0 from 402 users |

| | |
|---|---|
| CVE Reference | CVE-2011-4701 |
| Vendor | *fa*, http://d.hatena.ne.jp/ofnhwx/ |
| Vendor Response | Null |

## 2 Description

*CallConfirm* exposes the following content provider in the AndroidManifest.xml file, which are not properly protected:

- `<provider android:name=".provider.AllowListProvider" android:authorities="jp.gr.java_conf.ofnhwx.CallConfirm.AllowList" />`
- `<provider android:name=".provider.BlockListProvider" android:authorities="jp.gr.java_conf.ofnhwx.CallConfirm.BlockList" />`
- `<provider android:name=".provider.EditNumberProvider" android:authorities="jp.gr.java_conf.ofnhwx.CallConfirm.EditNumber" />`

Through the provider jp.gr.java_conf.ofnhwx.CallConfirm.AllowList and jp.gr.java_conf.ofnhwx.CallConfirm.BlockList, a malicious application on the same device can query, modify and delete the allow list and block list set by the user.

*** authors with equal contributions**

# 3 Impact

This vulnerability enables an adversary to access and manipulate the allow list and block list set by the user, as shown in Figure 3. By doing so, an attacker can delete a record so that spam could come in or add a record to launching the DoS attack.

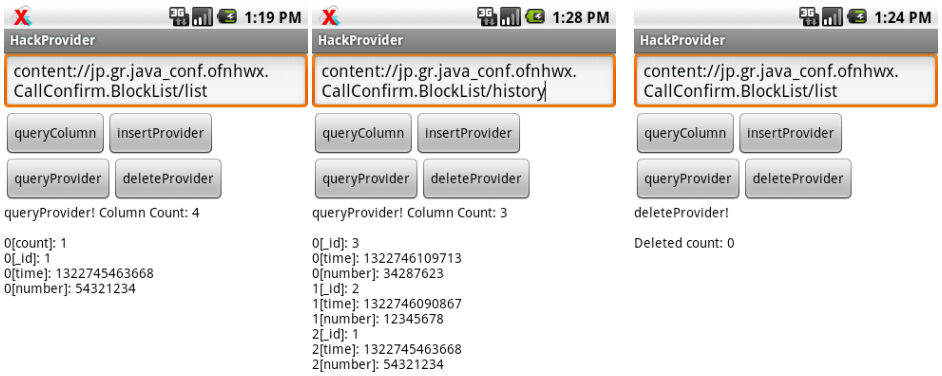


**Figure 1: Access and manipulate the user's block list.**

# 4 Solution

We are trying our best to contact *fa* 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 vulnerable content provider and the corresponding table are listed as follows:

| Content Provider Authority | Table Name |
|---|---|
| **jp.gr.java_conf.ofnhwx.CallConfirm.AllowList** | list |
| **jp.gr.java_conf.ofnhwx.CallConfirm.BlockList** | list |
| **jp.gr.java_conf.ofnhwx.CallConfirm.BlockList** | history |
| **jp.gr.java_conf.ofnhwx.CallConfirm.EditNumber** | list |

**Sample attack codes:**

```
providerUri = Uri.parse("content://
jp.gr.java_conf.ofnhwx.CallConfirm.BlockList/list")
ContentResolver cr  = this.getContentResolver();


//Delete BlockList
```

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

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

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