

Design: Stealing Location in Android

Luoyin Feng

Information Security
Northeastern University(CN)
Shenyang, China
fengluoyin@gmail.com

Abstract—In this paper, the design of the app is discussed in detail. In general, this app functions as a dictionary that require permission: ACCESS_WIFI_STATE, CHANGE_WIFI_STATE, and INTERNET. However, it can locate the approximate location of the user. This goal is achieved by mainly three steps:(1) Know your Aps: Get the Aps' SSID, BSID and other information (2) Write a stealth user tracking app: this app should not be detected by user (3) Send the tracking data off the device: send the email secretly to access the location of the user.

Keywords—Android, Mobile Security, Tracking.

I. INTRODUCTION

Android has been gaining its popularity in recent years. The security mechanism of Android is of great importance. However, various kinds of vulnerabilities are exploited by the attackers that design and implement the malicious software to steal user information. Attacker is able to steal the user's location information and send it to a third party via E-mail. Therefore, I designed this app to locate the position of the user. To make it less possible for the user to be aware of this, I make it unable to interact with this app. Finally, if the user goes to the specific area, this app will send E-mail to my Gmail.

II. DESCRIPTIONS

I follow the Instruction of the website and get to know the mood of that "friend", so the first thing I do is to view the whole process and make every step clear. There are two main things need to be solved, one is how to locate the position when we get the Aps, the other one is how to make the app unable to see.

The key to stealing the location information lies in two steps: 1) Know your Aps. 2) Write a stealth user tracking app.3) Send the tracking data off the device.

The first step is to make a list and make it clear to use these Aps. I went to the WiGLE.net [1] which is a website that collects the Aps information all over the world. In this way we get the information about the Aps in INI and Hamersschlag in the period of 2014-2015. This list contains the SSID and BSSID of an Ap. Because as the website said that the dynamic way is not stable, they may change it often, so I choose to make a list. When I have this list, I begin to figure out how to use this list to locate one's position. After I search on the developer.android.com I found that I can use the WifiManager to scan the surround Aps. [2] Then I designed my own class WifiAdmin which is in charge of controlling the information of the Ap. The list is showed as Figure 1.

```
listResultofHam.add(new Scandata("00:1a:1e:8d:d1:83", "eduroam"));+
listResultofHam.add(new Scandata("d8:c7:c8:d4:0b:83", "eduroam"));+
listResultofHam.add(new Scandata("00:1a:1e:93:5e:e1", "CMU-GUEST"));+
listResultofHam.add(new Scandata("00:1A:1e:93:5e:e2", "eduroam"));+
+
+
listResultofINI.add(new Scandata("00:1c:b0:e9:69:10", "xineza"));+
listResultofINI.add(new Scandata("00:18:f8:58:c5:df", "CMU"));+
listResultofINI.add(new Scandata("00:1a:c1:38:78:ac", "Testworks"));+
listResultofINI.add(new Scandata("00:13:7f:33:33:90", "CMU"));+
listResultofINI.add(new Scandata("00:14:1b:5a:30:21", "PSC"));+
listResultofINI.add(new Scandata("00:18:84:12:75:8a", "Dancing Kirbys"));+
listResultofINI.add(new Scandata("00:13:10:35:69:6d", "ravicmu"));+
listResultofINI.add(new Scandata("00:1c:10:03:b5:09", "HalleEnd"));+
listResultofINI.add(new Scandata("00:18:74:09:69:81", "CMU-TP4"));+
```

Figure 1. list of Aps

The second step is to hide the app. At the beginning I thought it may need to change the system file. After searching on the internet, I found that in the manifest file of the app project, if I remove the LAUNCHER the app won't show in the menu. That is to say I can finish it in this way. At the same time, it requires that the app can run no matter whether the screen is locked, so that means I need to make this scan as a service which can run even the screen is locked. In this I set the interval time as 30 mins which means I will get an E-mail no matter if the user is in the specific area.

The third step is to send the information off the device. After setting the Gmail allowing less secure apps and smtp service and in this process I use the jar offered by course. [3] I make the app can send the information successfully. The information we need is listed in the Figure 2 and Figure 3.

This email was sent from an NEXUS 5, without user interaction.
Sent to your address to be awesome...

NO.1 :301->d8:42:ac:84:e9:9d->[WPA-PSK-CCMP+TKIP]
[WPA2-PSK-CCMP+TKIP][ESS]->2472->-50->0

NO.1 :301->d8:42:ac:84:e9:9d->[WPA-PSK-CCMP+TKIP]
[WPA2-PSK-CCMP+TKIP][ESS]->2472->-50->0

he is around Ham!!!!!!!!!!!!!!

he is around INI!!!!!!!!!!!!!!

Figure 2. the information of the Ap


Awesome HW3 Stuff!! ☆
From: flyfollowing <flyfollowing@gmail.com> 
Date: Sunday, Oct 11, 2015 3:09 PM
To: 448501751 <448501751@qq.com>

Figure 3. sender and title

III. ASSUMPTIONS

The TA left his phone laying around and not locked.

if the app is suspended in the background, that the user will not check the task manager

There is no anti-virus or anti-malware app installed on the phone.

The Aps information is not changed.

IV. EXPLANATIONS

A. *How to steal information about location*

Compare the surrounded Aps information with the Aps that we have already the location. If they are matched we can say

that the user is around the Ap and because the BSSID is unique so it can't be used for this process.

B. *How to hide the app in the launcher*

By removing the `<category android:name= "android.intent.category.LAUNCHER" />`

IV. CONCLUSION

This design of our app could help us implement the tracking from others secretly.

[1] <http://WiGLE.net>.

[2] <http://developer.android.com/reference/android>

[3] <http://mews.sv.cmu.edu/teaching/14829/f15/hw3.html>