Short Paper:
WifiLeaks: Underestimated Privacy Implications of the ACCESS_WIFI_STATE Android Permission

Jagdish Prasad Achara, Mathieu Cunche, Vincent Roca, and Aurélien Francillon

WiSec'14, Oxford, UK

July 25th, 2014
Android Permission System

- Location
- Internet
- Accounts
- Contacts

etc.
Android Permission System

Application

permissions

Location
Internet
Accounts
Contacts

etc.

(Nature-based classification)

Normal
Dangerous

(Protection level-based classification)
Android Permission System

145 Permissions

Application

permissions

Accounts

Contacts

e tc.
Android Permission System

145 Permissions

Network
- Internet
- Access_wifi
- etc.

Location
- Fine_location
- Mock_location
- etc.
- etc.

(Accounts, Contacts)

(Nature-based classification)
Android Permission System

145 Permissions

Network
- Internet
- Access_wifi
- etc.

Location
- Fine_location
- Mock_location
- etc.

Accounts

Contacts

Dangerous
- Fine_location
- Coarse_location
- etc.

Normal
- Access_wifi
- Access_network
- etc.

permissions

(Nature-based classification)  (Protection level-based classification)

etc.

etc.
Effectiveness of Android Permission System

- **Poor understanding** [Felt et. al. SOUPS’12]

- **Private Information retrieval without any permission** [Zhou et. al. CCS’13]
  - Privateae Information: Geolocation, Identity etc.


Effectiveness of Android Permission System

- **Poor understanding** [Felt et. al. SOUPS’12]

- **Private Information retrieval without any permission** [Zhou et. al. CCS’13]
  - Private Information: Geolocation, Identity etc.


The case of ACCESS_WIFI_STATE permission (1)

Permission description displayed to users

- Required to access raw Wi-Fi data
- Group [2]: ‘Network’
- Protection level [1]: ‘Normal’

Looks innocuous at first glance!

The case of ACCESS_WIFI_STATE permission (2)

In fact, it looks innocuous but it is not!

It is known that:

- **Raw Wi-Fi data**: A source of sensitive information
  1. Surrounding Wi-Fi APs $\rightarrow$ Approximate user location
  2. Wi-Fi MAC address $\rightarrow$ A unique device identifier
  3. Configured Wi-Fi APs $\rightarrow$ Travel history and Social links [1]
  4. Connected Wi-Fi APs and time $\rightarrow$ Points of interests

The case of ACCESS_WIFI_STATE permission (2)

In fact, it looks innocuous but it is not!

It is known that:

- **Raw Wi-Fi data**: A source of sensitive information
  1. Surrounding Wi-Fi APs $\rightarrow$ Approximate user location
  2. Wi-Fi MAC address $\rightarrow$ A unique device identifier
  3. Configured Wi-Fi APs $\rightarrow$ Travel history and Social links [1]
  4. Connected Wi-Fi APs and time $\rightarrow$ Points of interests

The case of ACCESS_WIFI_STATE permission (2)

In fact, it looks innocuous but it is not!

It is known that:

- **Raw Wi-Fi data**: A source of sensitive information
  1. Surrounding Wi-Fi APs → Approximate user location
  2. Wi-Fi MAC address → A unique device identifier
  3. Configured Wi-Fi APs → Travel history and Social links [1]
  4. Connected Wi-Fi APs and time → Points of interests

The case of ACCESS_WIFI_STATE permission (2)

In fact, it looks innocuous but it is not!

It is known that:

- **Raw Wi-Fi data**: A source of sensitive information
  - **1** Surrounding Wi-Fi APs → Approximate user location
  - **2** Wi-Fi MAC address → A unique device identifier
  - **3** Configured Wi-Fi APs → Travel history and Social links [1]
  - **4** Connected Wi-Fi APs and time → Points of interests

The case of ACCESS_WIFI_STATE permission (2)

In fact, it looks innocuous but it is not!

It is known that:

- **Raw Wi-Fi data**: A source of sensitive information
  - 1. Surrounding Wi-Fi APs → Approximate user location
  - 2. Wi-Fi MAC address → A unique device identifier
  - 3. Configured Wi-Fi APs → Travel history and Social links [1]
  - 4. Connected Wi-Fi APs and time → Points of interests

Motivation/Goals

As this permission seems exploitable, two questions raised:

1. Do users understand the implications of this permission?
   • i.e., what is the user perception of this permission?

2. Is this permission already being exploited by Apps?
   • i.e., what is the current situation on Google PlayStore?
Motivation/Goals

As this permission seems exploitable, two questions raised:

1. Do users understand the implications of this permission?
   • i.e., what is the user perception of this permission?

2. Is this permission already being exploited by Apps?
   • i.e., what is the current situation on Google PlayStore?
Survey Description

• A total of 156 users answered

• Diffused through social media and mailing-lists

• Composed of 12 questions divided into 3 parts:
  1. Demographic info
  2. User attitude towards privacy and his experience on Android
  3. User perception of the ACCESS_WIFI_STATE permission
A digest of Survey Results

1. **Less risky** than other permissions (like Geoloc)!

2. Privacy implications (geolocation, travel history) are not well understood
   - Less than half know about geolocalization!
   - Less than half know about device unique identifier!
   - Only 35% know about previously visited locations!
A digest of Survey Results

1. Less risky than other permissions (like Geoloc)!
2. Privacy implications (geolocation, travel history) are not well understood
   - Less than half know about geolocalization!
   - Less than half know about device unique identifier!
   - Only 35% know about previously visited locations!
Application Analysis: Overview

**First Step:** Permission analysis through crawling [1]:

- # of Apps: 2700 Apps (100 * 27 categories)
- Results: 41% Apps request ACCESS_WIFI_STATE

**Second Step:** 998 APKs requesting this permission are downloaded for:

1. Static analysis
2. Dynamic analysis (only 88 Apps are chosen based on static analysis results)

Application Analysis: Overview

**First Step**: Permission analysis through crawling [1]:
- # of Apps: 2700 Apps (100 * 27 categories)
- Results: 41% Apps request ACCESS_WIFI_STATE

**Second Step**: 998 APKs requesting this permission are downloaded for:
1. Static analysis
2. Dynamic analysis (only 88 Apps are chosen based on static analysis results)

Application Analysis: Overview

**First Step:** Permission analysis through crawling [1]:
- # of Apps: 2700 Apps (100 * 27 categories)
- Results: 41% Apps request ACCESS_WIFI_STATE

**Second Step:** 998 APKs requesting this permission are downloaded for:
1. Static analysis
2. Dynamic analysis (only 88 Apps are chosen based on static analysis results)

Static Analysis: Technique

- Custom tool (on top of Androguard [1])
- Analysis: Methods of WifiManager class [2]
- 3 privacy-sensitive methods:
  1. `getScanResults()`: List of surrounding Wi-Fi APs
  2. `getConnectionInfo()`: Connected AP Info + Wi-Fi MAC
  3. `getConfiguredNetworks()`: SSIDs of previously connected APs

Static Analysis: Technique

- Custom tool (on top of Androguard [1])
- Analysis: Methods of WifiManager class [2]
  - 3 privacy-sensitive methods:
    1. `getScanResults()`: List of surrounding Wi-Fi APs
    2. `getConnectionInfo()`: Connected AP Info + Wi-Fi MAC
    3. `getConfiguredNetworks()`: SSIDs of previously connected APs

Static Analysis: Technique

- Custom tool (on top of Androguard [1])
- Analysis: Methods of WifiManager class [2]
- 3 privacy-sensitive methods:
  1. `getScanResults()`: List of surrounding Wi-Fi APs
  2. `getConnectionInfo()`: Connected AP Info + Wi-Fi MAC
  3. `getConfiguredNetworks()`: SSIDs of previously connected APs

Static Analysis: Technique

- Custom tool (on top of Androguard [1])
- Analysis: Methods of WifiManager class [2]
- 3 privacy-sensitive methods:
  1. `getScanResults()`: List of surrounding Wi-Fi APs
  2. `getConnectionInfo()`: Connected AP Info + Wi-Fi MAC
  3. `getConfiguredNetworks()`: SSIDs of previously connected APs

Static Analysis: Technique

- Custom tool (on top of Androguard [1])
- Analysis: Methods of WifiManager class [2]
- 3 privacy-sensitive methods:
  1. `getScanResults()`: List of surrounding Wi-Fi APs
  2. `getConnectionInfo()`: Connected AP Info + Wi-Fi MAC
  3. `getConfiguredNetworks()`: SSIDs of previously connected APs

Static Analysis: Technique

- Custom tool (on top of Androguard [1])
- Analysis: Methods of WifiManager class [2]
- 3 privacy-sensitive methods:
  1. `getScanResults()`: List of surrounding Wi-Fi APs
  2. `getConnectionInfo()`: Connected AP Info + Wi-Fi MAC
  3. `getConfiguredNetworks()`: SSIDs of previously connected APs

---

Static Analysis: Results

App category wise distribution
Static Analysis: Results

App category wise distribution
Static Analysis: Results

<table>
<thead>
<tr>
<th>ConnectionInfo</th>
<th>ScanResults</th>
<th>ConfiguredNetworks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Third-party</td>
<td># Apps</td>
<td>Third-party</td>
</tr>
<tr>
<td>inmobi.com</td>
<td>74</td>
<td>inmobi.com</td>
</tr>
<tr>
<td>chartboost.com</td>
<td>55</td>
<td>domob.cn</td>
</tr>
<tr>
<td>tapjoy.com</td>
<td>49</td>
<td>mologiq.com</td>
</tr>
<tr>
<td>vungle.com</td>
<td>47</td>
<td>tencent.com</td>
</tr>
<tr>
<td>jirbo.com</td>
<td>43</td>
<td>skyhookwireless.com</td>
</tr>
</tbody>
</table>

**Top 5 third-parties accessing various methods**

Notions adopted:

- **First-party**: App developer,
  **Third-party**: Included libraries

- class_package_name $\neq$ main_package_name $\implies$ third_party
Dynamic Analysis: Technique

- Modification of Android OS to log interesting events...
- The modification includes methods from:
  1. WiFiManager and WifiInfo class
  2. Network stack (clear-text or ssl)
  3. Data modification APIs (hashes and encryption)
- Logged events are further analyzed automatically
Dynamic Analysis: Results

<table>
<thead>
<tr>
<th>Info</th>
<th>Third-parties</th>
<th>First-parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC Address</td>
<td>appsflyer.com (SSL), revmob.com (SSL), adsmogo.mobi (plain-text), adsmogo.org (plain-text), vungle.com (plain-text), supersonicads.com (plain-text), trademob.net (SSL), sponsorpay.com (SSL), beintoo.com (SSL), adsmogo.com (plain-text), 115.182.31.2/3/4 (plain-text), tapjoyads.com (SSL)</td>
<td>Not found</td>
</tr>
<tr>
<td>(B)SSID of connected AP</td>
<td>inmobi.com (SSL), 93.184.219.82 (plain-text)</td>
<td>Not found</td>
</tr>
<tr>
<td>Wi-Fi Scan Info</td>
<td>inmobi.com (SSL), fastly.net (SSL)</td>
<td>badoo.com (SSL), foursquare.com(SSL)</td>
</tr>
</tbody>
</table>
Dynamic Analysis: Results

<table>
<thead>
<tr>
<th>Info</th>
<th>Third-parties</th>
<th>First-parties</th>
<th>ACCESS_NETWORK_STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC Address</td>
<td>appsflyer.com (SSL), revmob.com (SSL), admogo.mobi (plain-text), admogo.org (plain-text), vungle.com (plain-text), supersonicads.com (plain-text), trademob.net (SSL), sponsorpay.com (SSL), beintoo.com (SSL), admogo.com (plain-text), 115.182.31.2/3/4 (plain-text), tapjoyads.com (SSL)</td>
<td>Not found</td>
<td>ACCESS_WIFI_STATE</td>
</tr>
<tr>
<td>(B)SSID of connected AP</td>
<td>inmobi.com (SSL), 93.184.219.82 (plain-text)</td>
<td>Not found</td>
<td>ACCESS_FINE_LOCATION</td>
</tr>
<tr>
<td>Wi-Fi Scan Info</td>
<td>inmobi.com (SSL), fastly.net (SSL)</td>
<td></td>
<td>ACCESS_FINE_LOCATION</td>
</tr>
</tbody>
</table>

Data collection and transmission to third-parties is a reality!

- MAC Address transmission to third-parties (even in CLEAR!)
Dynamic Analysis: Results

<table>
<thead>
<tr>
<th>Info</th>
<th>Third-parties</th>
<th>First-parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC Address</td>
<td>appsflyer.com (SSL), revmob.com (SSL), adsmogo.mobi (plain-text), adsmogo.org (plain-text), vungle.com (plain-text), supersonicads.com (plain-text), trademob.net (SSL), sponsorpay.com (SSL), beintoo.com (SSL), adsmogo.com (plain-text), 115.182.31.2/3/4 (plain-text), tapjoyads.com (SSL)</td>
<td>Not found</td>
</tr>
<tr>
<td>(B)SSID of connected AP</td>
<td>inmobi.com (SSL), 93.184.219.82 (plain-text)</td>
<td>Not found</td>
</tr>
<tr>
<td>Wi-Fi Scan Info</td>
<td>inmobi.com (SSL), fastly.net (SSL)</td>
<td>badoo.com (SSL), foursquare.com(SSL)</td>
</tr>
</tbody>
</table>

Data collection and transmission to third-parties is a reality!

- MAC Address transmission to third-parties (even in CLEAR!)
- Wi-Fi Scan info transmission to both first and third-parties
Dynamic Analysis: Results

<table>
<thead>
<tr>
<th>Info</th>
<th>Third-parties</th>
<th>First-parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC Address</td>
<td>appsflyer.com (SSL), revmob.com (SSL), adsmogo.mobi (plain-text), adsmogo.org (plain-text), vungle.com (plain-text), supersonicads.com (plain-text), trademob.net (SSL), sponsorpay.com (SSL), beintoo.com (SSL), adsmogo.com (plain-text), 115.182.31.2/3/4 (plain-text)⁷, tapjoyads.com (SSL)</td>
<td>Not found</td>
</tr>
<tr>
<td>(B)SSID of connected AP</td>
<td>inmobi.com (SSL), 93.184.219.82 (plain-text)</td>
<td>Not found</td>
</tr>
<tr>
<td>Wi-Fi Scan Info</td>
<td>inmobi.com (SSL), fastly.net (SSL)</td>
<td>badoo.com (SSL), foursquare.com(SSL)</td>
</tr>
</tbody>
</table>

Data collection and transmission to third-parties is a reality!

- MAC Address transmission to third-parties (even in CLEAR!)
- Wi-Fi Scan info transmission to both first and third-parties

What if I turn off my location to all Apps? → Out of luck!
Potential Solution

1. Protection of Wi-Fi scan results with location permissions
   - It is currently the case with neighboring cell towers
Potential Solution

1. Protection of Wi-Fi scan results with location permissions
   - It is currently the case with neighboring cell towers

2. Change of protection level: From ‘Normal’ to ‘Dangerous’
Potential Solution

1. Protection of Wi-Fi scan results with location permissions
   - It is currently the case with neighboring cell towers

2. Change of protection level: From ‘Normal’ to ‘Dangerous’

3. Modification of Permission description
   - Proposal for Improvement: “Allows the app to view information about Wi-Fi networking. MAC address can be used for user tracking and the list of configured Wi-Fi APs may reveal travel history.”
Conclusion

- **ACCESS_WIFI_STATE permission**: A source of various user PII

- **Privacy implications of the permission are not well understood**
Conclusion

• ACCESS_WIFI_STATE permission: A source of various user PII

• Privacy implications of the permission are not well understood

• 41% applications request this permission

• Permission exploitation already started:
  • Getting user location without dedicated location permissions
  • For tracking purposes
  • To know users’ points of interests
Conclusion

• ACCESS_WIFI_STATE permission: A source of various user PII

• Privacy implications of the permission are not well understood

• 41% applications request this permission

• Permission exploitation already started:
  • Getting user location without dedicated location permissions
  • For tracking purposes
  • To know users’ points of interests

Solution exists!
Thanks for your attention!

Questions?