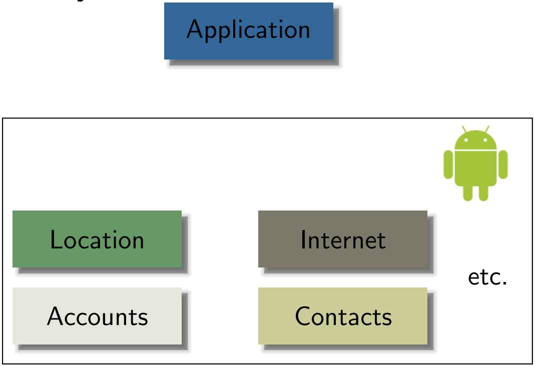
Short Paper: WifiLeaks: Underestimated Privacy Implications of the ACCESS_WIFI_STATE Android Permission

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

WiSec'14, Oxford, UK

July 25th, 2014

Android Permission System



Application

Location

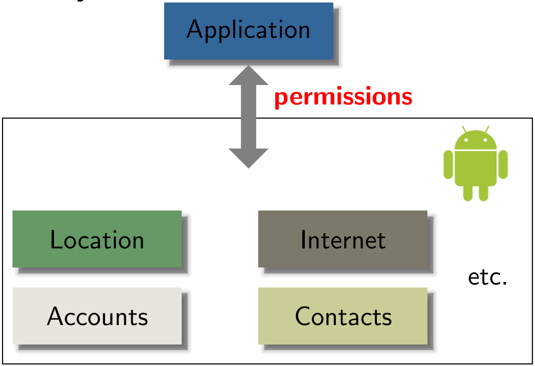
Internet

Accounts

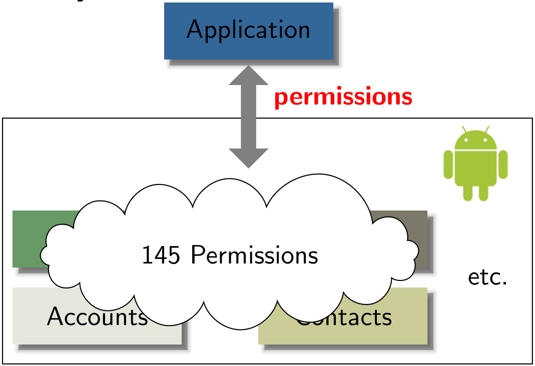
Contacts

etc.

Android Permission System



Android Permission System



Application

permissions

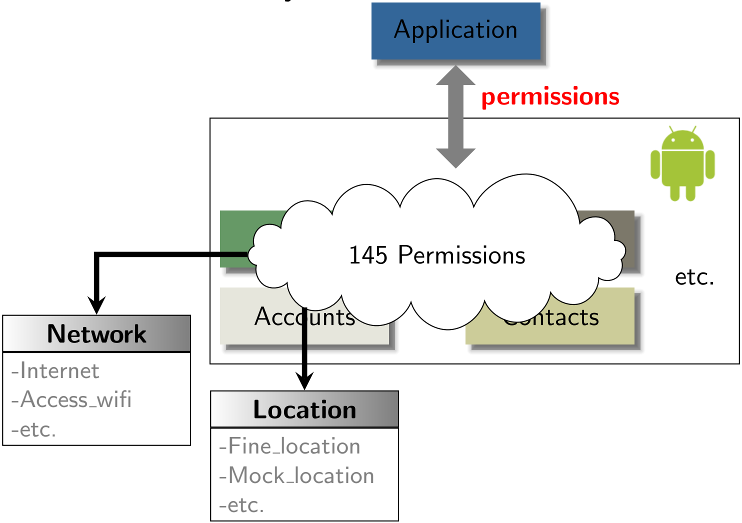
145 Permissions

Accounts

Contacts

etc.

Android Permission System



Application

permissions

145 Permissions

Accounts

Contacts

etc.

Network

-Internet
-Access_wifi
-etc.

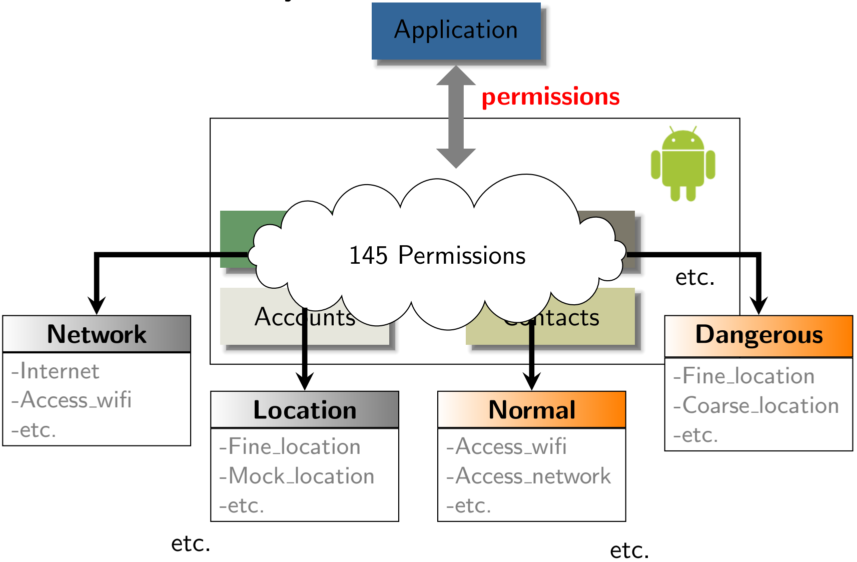
Location

-Fine_location
-Mock_location
-etc.

etc.

(Nature-based classification)

Android Permission System



Application

permissions

145 Permissions

Accounts

Contacts

etc.

Network

-Internet
-Access_wifi
-etc.

Location

-Fine_location
-Mock_location
-etc.

Normal

-Access_wifi
-Access_network
-etc.

Dangerous

-Fine_location
-Coarse_location
-etc.

etc.

etc.

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

Effectiveness of Android Permission System

- Poor understanding [Felt et. al. SOUPS'12]
- Private Information retrieval without any permission [Zhou et. al. CCS'13]
    - Privatae Information: Geolocation, Identity etc.

[Felt et. al. SOUPS'12] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner. Android permissions: User attention, comprehension, and behavior. SOUPS '12, New York, NY, USA, 2012. ACM.

[Zhou et. al. CCS'13] X. Zhou, S. Demetriou, D. He, M. Naveed, X. Pan, X. Wang, C. A. Gunter, and K. Nahrstedt. Identity, location, disease and more: Inferring your secrets from android public resources. In ACM CCS 2013.

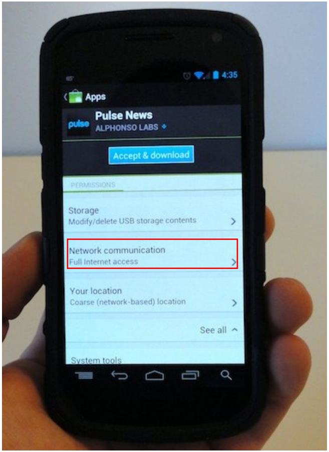
Effectiveness of Android Permission System

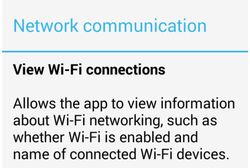
- Poor understanding [Felt et. al. SOUPS'12]
- Private Information retrieval without any permission [Zhou et. al. CCS'13]
    - Privatae Information: **Geolocation**, Identity etc.

[Felt et. al. SOUPS'12] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner. Android permissions: User attention, comprehension, and behavior. SOUPS '12, New York, NY, USA, 2012. ACM.

[Zhou et. al. CCS'13] X. Zhou, S. Demetriou, D. He, M. Naveed, X. Pan, X. Wang, C. A. Gunter, and K. Nahrstedt. Identity, location, disease and more: Inferring your secrets from android public resources. In ACM CCS 2013.

The case of ACCESS_WIFI_STATE permission (1)





Network communication

View Wi-Fi connections

Allows the app to view information about Wi-Fi networking, such as whether Wi-Fi is enabled and name of connected Wi-Fi devices.

Permission description displayed to users

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

Looks innocuous at first glance!

[1] http://developer.android.com/reference/android/Manifest.permission_group.html
[2] http://developer.android.com/guide/topics/manifest/permission-element.html

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

[1] M. Cunche, M.-A. Kaafar, and R. Boreli. Linking wireless devices using information contained in wi-fi probe requests. Pervasive and Mobile Computing, 2013.

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

[1] M. Cunche, M.-A. Kaafar, and R. Boreli. Linking wireless devices using information contained in wi-fi probe requests. Pervasive and Mobile Computing, 2013.

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

[1] M. Cunche, M.-A. Kaafar, and R. Boreli. Linking wireless devices using information contained in wi-fi probe requests. Pervasive and Mobile Computing, 2013.

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

[1] M. Cunche, M.-A. Kaafar, and R. Boreli. Linking wireless devices using information contained in wi-fi probe requests. Pervasive and Mobile Computing, 2013.

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

[1] M. Cunche, M.-A. Kaafar, and R. Boreli. Linking wireless devices using information contained in wi-fi probe requests. Pervasive and Mobile Computing, 2013.

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:

❶ Do users understand the implications of this permission?

- i.e., what is the user perception of this permission?

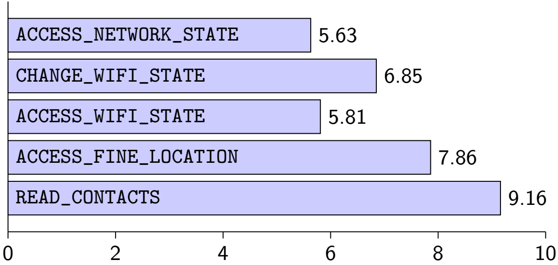
❷ Is this permission already being exploited by Apps?

- i.e., what is the current situation on Google PlayStore?

Survey Description

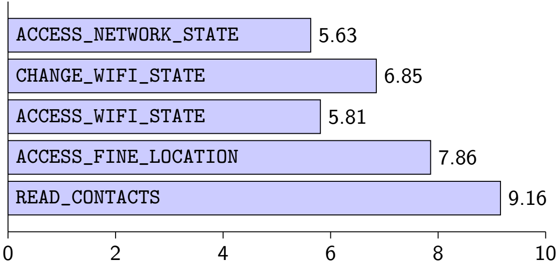
- A total of 156 users answered
- Diffused through social media and mailing-lists
- Composed of 12 questions divided into 3 parts:
  1. Demographic info
  2. User attitude towards privacy and his experience on Android
  3. User perception of the ACCESS_WIFI_STATE permission

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)

[1] https://github.com/egirault/googleplay-api

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)

[1] https://github.com/egirault/googleplay-api

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)

[1] https://github.com/egirault/googleplay-api

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

[1] https://code.google.com/p/androguard/
[2] http://developer.android.com/reference/android/net/wifi/WifiManager.html

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

[1] https://code.google.com/p/androguard/
[2] http://developer.android.com/reference/android/net/wifi/WifiManager.html

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

[1] https://code.google.com/p/androguard/
[2] http://developer.android.com/reference/android/net/wifi/WifiManager.html

Static Analysis: Technique

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

[1] https://code.google.com/p/androguard/

[2] http://developer.android.com/reference/android/net/wifi/WifiManager.html

Static Analysis: Technique

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

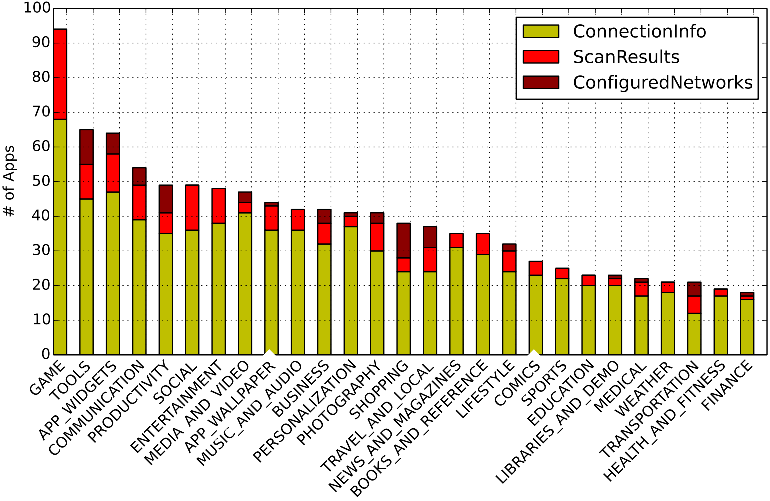
[1] `https://code.google.com/p/androguard/`
[2] `http://developer.android.com/reference/android/net/wifi/WifiManager.html`

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

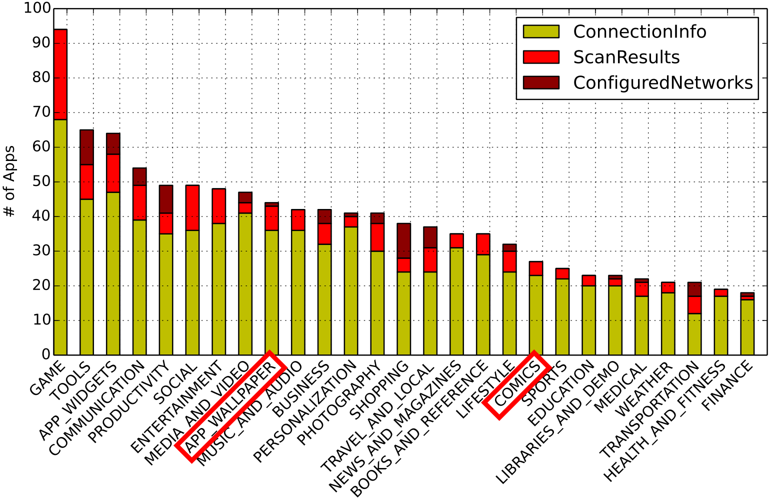
[1] https://code.google.com/p/androguard/
[2] http://developer.android.com/reference/android/net/wifi/WifiManager.html

Static Analysis: Results



App category wise distribution

Static Analysis: Results



App category wise distribution

Static Analysis: Results

ConnectionInfo		ScanResults		ConfiguredNetworks	
Third-party	**# Apps**	**Third-party**	**# Apps**	**Third-party**	**# Apps**
inmobi.com	74	inmobi.com	9	google.com	10
chartboost.com	55	domob.cn	9	mobiletag.com	4
tapjoy.com	49	mologiq.com	6	lechucksoftware.com	2
vungle.com	47	tencent.com	5	android.com	2
jirbo.com	43	skyhookwireless.com	4	Unibail.com	1

Top 5 third-parties accessing various methods

Notions adopted:

- First-party: App developer,
  Third-party: Included libraries
- class_package_name != main_package_name ⟹ 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

Info	Third-parties	First-parties
MAC Address	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)	Not found
(B)SSID of connected AP	inmobi.com (SSL), 93.184.219.82 (plain-text)	Not found
Wi-Fi Scan Info	inmobi.com (SSL), fastly.net (SSL)	badoo.com (SSL), foursquare.com(SSL)

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

Dynamic Analysis: Results

Info	Third-parties	First-parties
MAC Address	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)	Not found
(B)SSID of connected AP	inmobi.com (SSL), 93.184.219.82 (plain-text)	Not found
Wi-Fi Scan Info	inmobi.com (SSL), fastly.net (SSL)	badoo.com (SSL), foursquare.com(SSL)

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

- MAC Address transmission to third-parties (even in CLEAR!)

Dynamic Analysis: Results

Info	Third-parties	First-parties
MAC Address	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)	Not found
(B)SSID of connected AP	inmobi.com (SSL), 93.184.219.82 (plain-text)	Not found
Wi-Fi Scan Info	inmobi.com (SSL), fastly.net (SSL)	badoo.com (SSL), foursquare.com(SSL)

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

Info	Third-parties	First-parties
MAC Address	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)	Not found
(B)SSID of connected AP	inmobi.com (SSL), 93.184.219.82 (plain-text)	Not found
Wi-Fi Scan Info	inmobi.com (SSL), fastly.net (SSL)	badoo.com (SSL), foursquare.com(SSL)

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

❶ 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?