

# Title: 50 Ways to Leak Your Data: An Exploration of Apps' Circumvention of the Android Permissions System.

**Introduction:** Modern smartphone platforms implement permission-based models to protect access to sensitive data and system resources. Apps can circumvent the permission model and gain access to protected data without user consent by using both covert and side channels. Side channels present in the implementation of the permission system allow apps to access protected data and system resources without permission, whereas covert channels enable communication between two colluding apps so that one app can share its permission protected data with another app lacking those permissions.

**Test Environment:** Our Experiment setup uses Hybrid analysis techniques (combination of both static and dynamic analysis techniques) to triage suspicious apps and analyse their behaviours in depth. Static analysis involves scanning the code for all possible combinations of execution flows to understand potential execution behaviours (the behaviours of interest may include various privacy violations). Dynamic analysis studies an executable by running it and auditing its runtime behaviour. Typically, dynamic analysis benefits from running the executable in a controlled environment, such as an instrumented mobile OS, to gain observations of an app's behaviour.

We used this testing environment to find evidence of covert and side channel usage in 252,864 versions of 88,113 different Android apps, all of them downloaded from the U.S. Google Play Store using a purpose-built Google Play scraper. We executed each app version individually on a physical mobile phone equipped with a customized operating system and network monitor. This testbed allows us to observe apps' runtime behaviours both at the OS and network levels. We can observe how apps request and access sensitive resources and their data sharing practices. Before running each app, we gather the permission protected identifiers and data. We then execute each app while collecting all of its network traffic. We apply a suite of decoding to the traffic flows and search for the permission protected data in the decoded traffic. Then, we group the suspect transmissions by the data type sent and the destination where it was sent, because we found that the same data-destination pair reflects the same underlying side or covert channel.

Finally, we fingerprint the apps and libraries found using covert- and side-channels to identify the static presence of the same code in other apps in our corpus. A fingerprint is any string constant, such as specific filename or error message, that can be used to statically analyse our corpus to determine if the same technique exists in other apps that did not get triggered during our dynamic analysis phase.

We wrote a Google Play Store scraper to download the most popular apps under each category. Because the popularity distribution of apps is long tailed, our analysis of the 88,113 most-popular apps is likely to cover most of the apps that people currently use. As developers tend to update their Android software to add new functionality or to patch bugs [64], these updates can also be used to introduce new side and covert channels. Therefore, it is important to examine different versions of the same app, because they may exhibit different behaviours.

**Result:** We have identified five different types of side and covert channels in use among the 88,113 different Android apps in our dataset. Following table summarizes our findings and reports the number of apps and third-party SDK that we find exploiting these vulnerabilities in our dynamic analysis.

Data Type	Permission	Purpose/Use	Subsection	No of Apps		No of SDKs		Channel Type	
				Dynamic	Static	Dynamic	Static	Covert	Side
IMEI	READ_PHONE_STATE	Persistent ID	4.1	13	159	2	2	2	0
Device MAC	ACCESS_NETWORK_STATE	Persistent ID	4.2	42	12,408	1	1	0	1
Email	GET_ACCOUNTS	Persistent ID	Not Found						
Phone Number	READ_PHONE_STATE	Persistent ID	Not Found						
SIM ID	READ_PHONE_STATE	Persistent ID	Not Found						
Router MAC	ACCESS_WIFI_STATE	Location Data	4.3	5	355	2	10	0	2
Router SSID	ACCESS_WIFI_STATE	Location Data	Not Found						
GPS	ACCESS_FINE_LOCATION	Location Data	4.4	1	1	0	0	0	1

**Conclusion:** Here we can conclude that Android permission system have some vulnerability and this type of system comprises privacy of all user data.

Name- Saeem Ansari.      UID-2019430001

Name- Ajay Singh.      UID-2019430012

Title: 50 Ways to Leak Your Data: An Exploration of Apps' Circumvention of the Android Permissions System.

Proceeding - 28th USENIX Security Symposium, August 14 to 16 2019, Santa clara ,CA,USA.