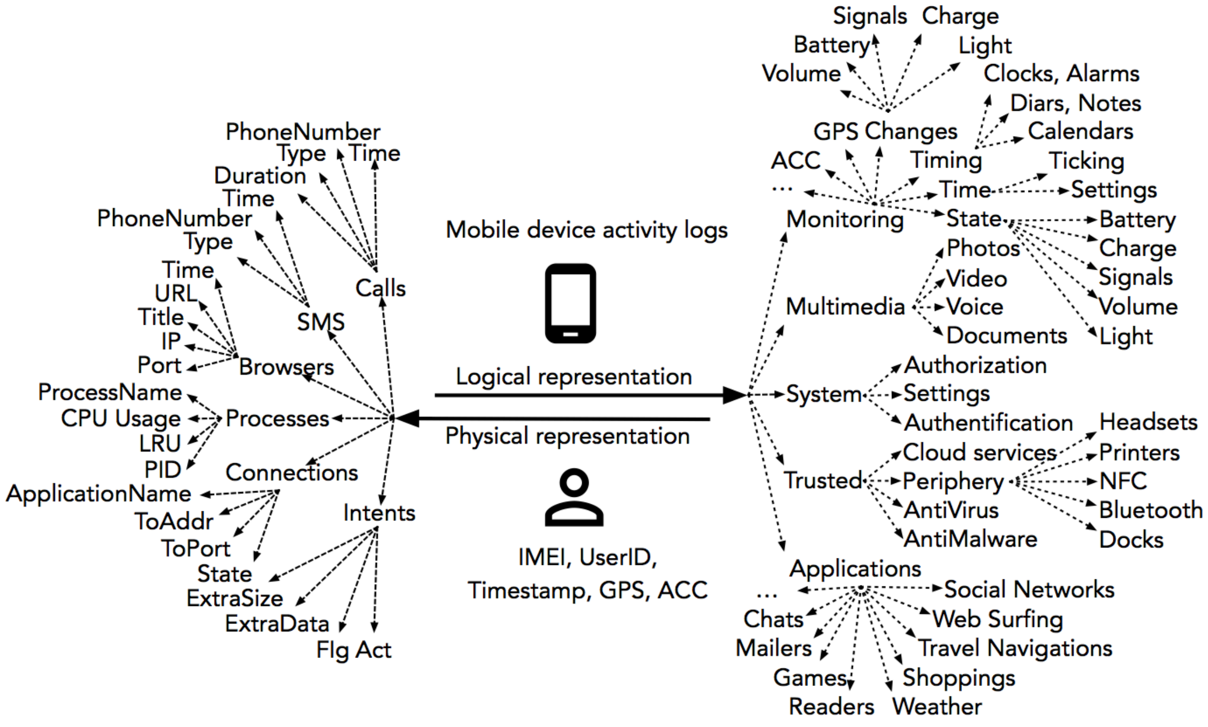
Log collection and Data features

All devices operating in the testbed are monitored in real-time and they transmit raw logs to the Logger server in predefined time period. Every mobile device is marked by their International Mobile Equipment Identity (IMEI) and can be dynamically located by positioning longitude and latitude values (via GPS). Thus, device movements can be described in the three-dimensional space with x, y, and z values coming from device accelerometers. Each mobile user is recognized via its userID. The userID can be considered to be equal to device's IMEI. Because of rather high expenses to equip volunteers with a larger number of devices, the data collection process is realized by our technological partners, who are also the owners of devices.

While our monitoring agents are deployed on the Android operating system, the intrusion activities are generated by modifications of the external metasploit library in order to investigate vulnerability Rapid7 [23]. The goal is to collect raw logs including threats for our work. Raw logs are recorded within device activities like: the history of calls, SMS and browser, intents, sampling of processes and connections. Logs collected from mobile devices can be categorized into groups according to the physical taxonomy as illustrated by Fig. 1 and the paragraphs below.

- Calls: Timestamp, Time, PhoneNum, Type, Duration ...
- SMS: Timestamp, Time, PhoneNum, Type, ...
- Browser history: Timestamp, Time, URL, Title, IP, Port, ...
- Processes: Timestamp, ProcessName, CPU usage, LRU (Least Recently Used), PID, ...
- Connections: Timestamp, Application, ToADDR, ToPort, FromADDR, FromPort, State, ...
- Intents: Timestamp, Act, Flg, ExtraSize, ExtraData, ...



Full names of intents, processes and connection types are available in Android Developers, [24] and in vendor technical manuals (Asus, Lenovo, LG, Samsung, and SonyEricson). Raw data (mobile device logs) belong to the human-generated data class, which has a high potential in volume, velocity, variety and veracity characteristics, as introduced earlier, but they are not so big as the machine-generated data. Just a half of gigabytes (GB) per mobile device per day without multimedia are gathered from twenty devices of volunteer users. Besides this, the data is also extremely noisy for the detection purpose. Two notable problems could be mentioned are over time duplicated information about processes as well as connections running on mobile devices, and missing values due to difficulties in the data collection phase in the real environment. From this viewpoint, to reduce features before analytics, all the physical raw logs are taxonomically mapped into logical classes, which cover monitoring, multimedia, system, trusted and user's applications. As presented above, the logical representation taxonomy of raw logs is expressed by Fig. 1.

Monitoring: information services include timing, signal and synchronizations (WiFi, Bluetooth, NFC), battery states, battery state changes, flashlights, power states, power state changes, charging, positioning (GPS), accelerometer values, background settings, the light intensity, volume, status monitoring and changes, time setting, time ticking, calendars, diaries, weather, hours, alarms, and so on;

- Multimedia: data generated from photos, video, voice (calls, music), text (SMS, documents, notes);
- System processes such as settings, authentication and authorization;
- Trusted applications are antivirus, antimalware, cloud and periphery services;
- User applications are used and installed based on user's demands for various purposes such as web surfing, sending and receiving emails, discussions, chats, social networking, collaborations, games, shopping, travel navigation, reading, and so on.

With the goal of detecting malware intrusions on mobile devices, we only focus on system and user application classes to collect and analyze logs. Main reason for the elimination is that log types monitored in other classes do not provide sensitive data for our detection purpose. User's activities on mobile phones are collected in two ways, as follows:

- Browser, call, SMS histories and intents are gathered gradually as soon as they appear on devices;
- Processes and connections are collected in the style of sampling in time periods, which are changed based on the stored data amount. Because applications and processes are sampled in predefined intervals (nearly per every minute), the log data contains a redundancy. Such repeated information arisen in sampling denotes that processes running on mobile operation system (OS) are dynamical. This phenomenon is expressed by Fig. 2.

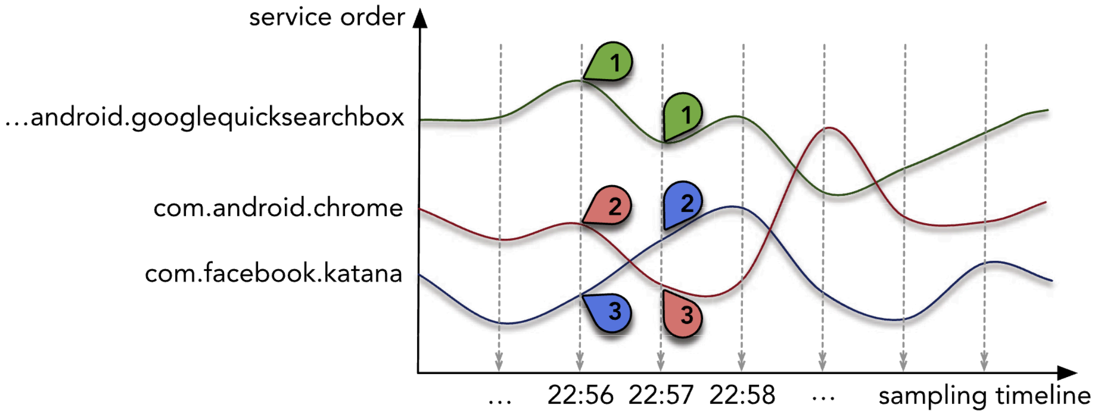


Fig. 2. Simplified illustration of OS service order for three processes in relation with sampling timeline.

Fig. 2 illustrates a simplified situation of OS service orders in relation to sampling timeline for three processes com.google.android.googlequicksearchbox, com.android.chrome and com.facebook.katana. It can be observed that at the time of 22:56, the sampled process order is search, chrome and facebook, from top to down. In the next sample time at 22:57, the order is search, facebook and chrome. Other observations can be made here consisting of almost all applications running on the devices as background processes without strictly start and stop states, where the relationship between each pair of applications, processes and connections are not of one-to-one correspondence. For instance, the application facebook owns several processes, namely com.facebook.katana, com.facebook.katana:dash, com.facebook.katana:nodex, com.facebook.orca and com.facebook.pages.app, but only processes com.facebook.katana, com.facebook.orca and com.facebook.pages.app relate to the web connection. This particular example illustrates the complexity of monitoring processes states within the testbed. The collected process logs represent only one of many data types, which are gathered during carrying out of our project.