February 2020

Limiting Application Access to Sensitive Data and Resources Through Use of a Sensitive Data Sandbox

Russell Quong

Limiting Application Access to Sensitive Data and Resources Through Use of a Sensitive Data Sandbox

Abstract:

This publication describes techniques and apparatuses to limit access by a software application to sensitive user data (*e.g.*, contacts, location, biometric data, photos). When a user installs or initially uses an application, the application may request permission to access user data, along with system resources (*e.g.*, global positioning satellite (GPS) data) and particular hardware (*e.g.*, microphones, cameras). Depending on the permission request, the user and the operating system (OS) may be unclear whether the data, resources, and/or hardware is needed by the application most of the time, some of the time, not needed at all, or whether the request is abusive. In some cases, the user is not able to provide partial access to the application. To protect the privacy of the user and minimize abuse, the OS utilizes a sensitive data sandbox to (a) evaluate access requests made by the application, (b) limit the frequency of requests, (c) limit the type and amount of user data and resources available within the sandbox, and (d) restrict the exporting of information from the sandbox.

Keywords:

Sandbox, isolated environment, sensitive information, personal information, data, sensitive data, user data, system resources, access, partial access, restricted access, permission request, application, privacy.

Background:

Operating system developers, software application developers, software application markets, and user equipment (UE) manufacturers increasingly offer users more product features and functionalities. As one example, a smartphone enables a user to connect to a variety of communication networks, make a phone call, text, email, participate in social networks, navigate, bank, shop, search for information, consume various types of media, play video games, and perform a variety of other functions. In addition, UE manufacturers integrate global navigation satellite system (GNSS) technology (*e.g.*, GPS), proximity sensors, biometric sensors, heart-rate sensors, cameras, microphones, and various other sensors in or on the UE to enhance user experience. The OS manages sensitive user data, such as contacts, photos, video, audio, location data, network connection data, user biometric data (*e.g.*, fingerprint data, voice-recognition data, facial recognition data), battery status, credit card information, user health and activity data, and other user-specific information, which may play a role in the functionality of various applications.

In aspects, applications run in a limited-access sandbox. When the user installs or initially uses an application, if the application needs to use resources or information outside of its sandbox, the application has to request the appropriate permission. For example, the application may request permission to access user data, UE resources (*e.g.*, GNSS location data, GPS location data), and/or particular hardware (*e.g.*, microphones, cameras, biometric sensors). Depending on the request, the OS or application market may prompt the user to approve the permission request.

As an example, Mary downloads an application called *Bill's Coffee* provided by her favorite chain of coffee shops. *Bill's Coffee* allows Mary to remotely order drinks from the nearest *Bill's Coffee* shop, pay with her phone, track her activity rewards, and find the nearest shop. As she initially uses the *Bill's Coffee* application, the OS prompts Mary to approve the permission

request to access her location data. Mary may understand why *Bill's Coffee* requests her location if she is searching for a local shop or placing a coffee order. She may be unclear, however, whether the application will continuously access her location data or abuse it to send her unsolicited advertisements and track her whereabouts. As a result, Mary may wish to grant *Bill's Coffee* partial access to her location data. The OS on Mary's phone, however, does not have the option to grant *Bill's Coffee* partial access, so Mary opts to grant the application with full access to her location data.

This example demonstrates how a user may be unsure whether they need to accept or deny a permission request because it is unclear whether the permission is required most of the time, some of the time, not needed at all, or whether the request is abusive. It also illustrates that there are times when the user should provide applications partial or limited access to sensitive user data (*e.g.*, health data, biometric data) and system resources (*e.g.*, GPS location).

It is desirable to have a technological solution that evaluates permission requests made by an application, limits the frequency of requests for sensitive user data and system resources, and restricts the information made available to applications. In addition, because there is an increasing risk of sensitive data abuse, it is desirable to use a secure sandbox to restrict access by an application to sensitive data.

Description:

This publication describes a sensitive data sandbox to restrict access to and use of sensitive user data (*e.g.*, calendar, photos, biometric data, credit card information) and UE resources (*e.g.*, GNSS data, GPS data, camera). When a user downloads or initially uses the application, the user may be prompted to grant the application permission to access user data and system resources. If

the user grants the permission request for sensitive user data or system resources, the OS evaluates the permission requests, limits the frequency and type of requests for user data or system resources, and restricts the information available to the application outside of the sandbox. In addition, the OS may employ different limits and restrictions based on the application and/or based on the type of user data accessed.

When the application requests permission, the user and OS may be unclear on how often the application accesses the requested data or resources and how it uses the data or resources. On the other hand, the application may not have the ability to provide certain desired features and functionalities without gaining access to sensitive user data and system resources.

A sensitive data sandbox controls and limits access by an application to sensitive data and system resources. As described herein, a data sandbox is an isolated environment within the UE that allows an application to access and process sensitive data and system resources under certain parameters to increase the security of the personal information of the user.

The sensitive data sandbox would allow an application to run a handler, which is generally a small amount of application code executing requested functions or methods, in the sandbox to access and process sensitive data and system resources. In other words, the application would still be able to access the desired data and resources, but it would be required to access these data and resources within the sensitive data sandbox. Exceptions, however, could be made for applications performing critical or time-sensitive functions. For example, a navigation application providing real-time directions may be given unrestricted, direct access to the user's current location. The sensitive data sandbox would also permit the handler to return responses based on or derived from the sensitive user data or system resources, but it would restrict the handler from exporting

sensitive data and system resource data itself. The OS ensures that an application does not have access to sensitive data except through the sandbox.

The sensitive data sandbox is illustrated by considering the *Bill's Coffee* application example described above. Mary has downloaded the application and granted the *Bill's Coffee* application with permission to access her location data. *Bill's Coffee* wants to know if Mary is near a *Bill's Coffee* shop so it can prompt her that she is near a shop or send her advertisements.

Without the described sensitive data sandbox, the *Bill's Coffee* application would be able to read Mary's current location L1 periodically. The application could then compute Mary's distance from all stores and determine whether Mary is currently within a half mile (or some other distance threshold) of one of its coffee shops. If Mary is within a half mile of a *Bill's Coffee* shop, then the application could send her a notification with the address of the nearest coffee shop and a coupon for a seasonal drink. *Bill's Coffee*, however, does not require Mary's actual location L1 (for example, the current longitude and latitude) to determine if Mary is near a coffee shop. The application only needs to determine whether Mary is within half a mile of one of its stores. By providing location data to *Bill's Coffee* upon request, there is a risk that the application may abuse this data by frequently accessing and storing it.

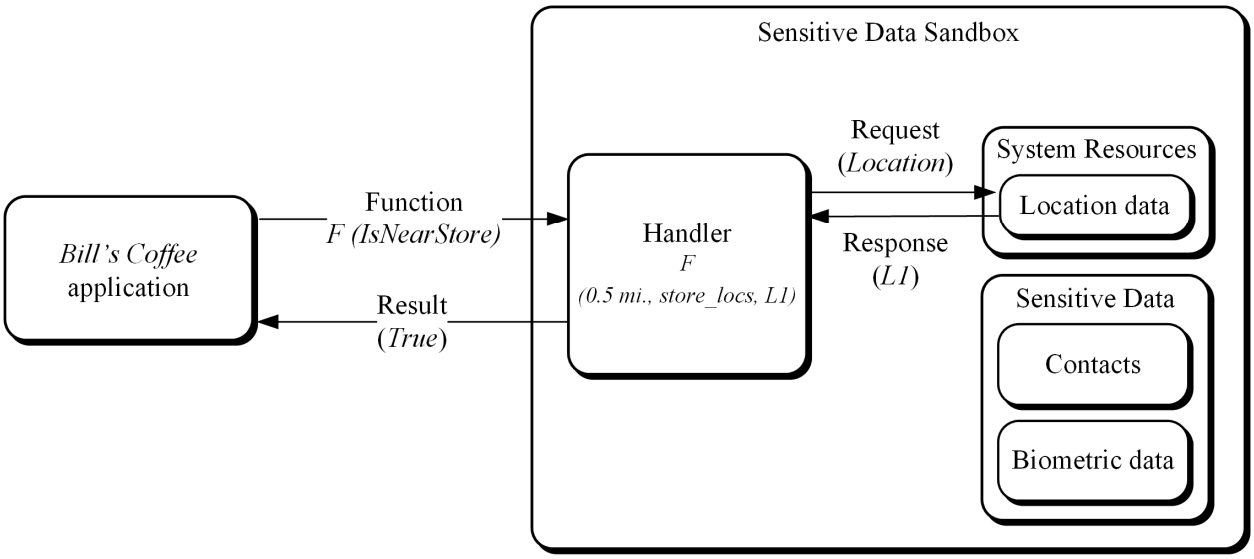

Figure 1

To demonstrate the sensitive data sandbox, consider the *Bill's Coffee* example illustrated in Figure 1. The application would define a function F (*e.g.*, IsNearStore) to determine whether Mary is within a half mile of a *Bill's Coffee* shop. Along with the function F, the application would also load location data (*e.g.*, store_locs) of all *Bill's Coffee* shops into the sandbox from the internal memory of Mary's phone or from an external networked location. The application handler would then execute function F in the sensitive data sandbox. The function F accesses Mary's current location L1 and performs a nearness check (IsNearStore) against the "store_locs" data provided by *Bill's Coffee*. The function F, however, only returns a "true" or "false" response; the handler cannot send Mary's location data L1 to the application. *Bill's Coffee* obtains the required information for its notification feature, but Mary's sensitive location data is kept secure and is not susceptible to abuse.

The sensitive data sandbox may utilize additional restrictions to improve the security of sensitive user data and prevent abuse. For example, the sandbox may limit how often an application may access the sandbox to retrieve certain types of sensitive data or system resources

per time interval. In the *Bill's Coffee* scenario, the sandbox could limit the application handler to one request of Mary's current location per hour.

As another parameter, the sandbox could limit the amount of information an application exports per time interval. In the *Bill's Coffee* scenario, the handler was permitted to return a single-bit of information, namely a "true" or "false" result for the function F based on Mary's current location L1. The application was not allowed to export the location data itself, which requires at least 40 bits of information. *Bill's Coffee*, however, could still attempt to approximate Mary's current location by iteratively asking if Mary is near each of its coffee shops. The sensitive data sandbox could limit the application to exporting two bits of information per hour to prevent *Bill's Coffee* from indirectly determining Mary's location. As a result, *Bill's Coffee* could only run function F (or a similar variant) twice per hour. A per-application and per-data-type restriction on the amount of information that returned from the sensitive data sandbox would prevent many types of abuse.

Additionally, the sandbox could limit the number of different parameters passed to the handler per time interval. In the *Bill's Coffee* scenario, the sandbox could prevent the handler from loading different location data sets. As a result, the function F would be restricted to performing a nearness check against the same "store_locs" data each time and would not be able to iteratively determine Mary's location. A limit on the parameters passed into the sandbox would prevent iterative attacks to obtain sensitive data.

Throughout this disclosure, examples are described where a computing system (*e.g.*, the UE, a client device, a server device, a computer, or another type of computing system) or application thereon may analyze information (*e.g.*, contacts, location, biometric data, and photos) associated with a user, such as the location data mentioned with respect to Figure 1. Further to the

descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, applications, and/or features described herein may enable collection of information (*e.g.*, information about a user's social network, social actions, social activities, profession, a user's preferences, a user's current location), and if the user is sent content or communications from a server. The computing system can be configured only to use the information after the computing system receives explicit permission from the user of the computing system to use the data. For example, in situations where an application analyzes location data to provide target advertisements, individual users may be provided with an opportunity to provide input to control whether an application can access and make use of the data. Further, individual users may have constant control over what applications can or cannot do with the information. In addition, information collected may be pre-treated in one or more ways before it is transferred, stored, or otherwise used, so that personally-identifiable information is removed. For example, a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level) so that a particular location of a user cannot be determined. Thus, the user may have control over whether information is collected about the user and the user's device, and how such information, if collected, may be used by the computing device, an application, and/or a remote computing system.

In conclusion, the sensitive data sandbox evaluates and restricts application access to and use of sensitive user data and system resources to avoid unintended and malicious abuse of sensitive information.

References:

[1] Patent Publication: US20190065680A1. Secure Computing Systems and Methods. Priority Date: May 1, 2014.

[2] Chen, Xin, Heqing Huang, Sencun Zhu, Qing Li, and Quanlong Guan. "SweetDroid: Toward a Context-Sensitive Privacy Policy Enforcement Framework for Android OS." *WPES '17: Proceedings of the 2017 on Workshop on Privacy in the Electronic Society*, October 30, 2017. https://doi.org/10.1145/3139550.3139552.