Technical Disclosure Commons

Defensive Publications Series

October 27, 2016

Automated Application Permissions Setting

Victor Carbune

Daniel Keysers

Thomas Deselaers

AUTOMATED APPLICATION PERMISSIONS SETTING

ABSTRACT

Operating systems of computing devices include permission management features to grant software applications (apps) access to various hardware and software components. Permissions may be configured using permission sets that each specify different levels of access. A user can specify the level of access to an app by selecting a permission set. A conventional permission set either grants or restricts access to a component. Techniques are described that provide selective access to a component by automatically inferring fine-grained permissions from various user-specific and other available signals.

KEYWORDS

- Mobile apps
- App permission
- Fine-grained access
- Permission predictor

BACKGROUND

An application (app) on a smartphone, tablet, personal computer, or other computing device seeks permission from a user to access various components. For example, an app might seek access to a camera, GPS (location information), the user’s contact list, and so on. Such permissions can be obtained during initial installation or subsequent use of the app. Apps typically seek component permissions in a coarse-grained manner, that is, the permission is requested to make use of the component, without substantial restrictions.

For example, if an app requests and obtains permission from the user to access photographs taken using a given device's camera, the app may obtain access to all photographs. There is no provision to specify, for example, that the app be allowed access only to a subset of photos that meet certain criteria, e.g., taken at a particular location, during a certain time period, etc. As a further example, if an app seeks permission to access a transducer, such as a speaker on the device, such permission is provided at all times and locations. Current permission management features do not allow setting fine-grained permissions, e.g., by indicating that the app be denied access to the speaker at particular locations such as an office.

Setting fine-grained permissions can impose a burden on the user, as these requests might be frequent and/or perceived to be time-consuming tasks that require user input.

DESCRIPTION

This disclosure describes techniques to infer and set fine-grained app permissions based upon a variety of signals, when a user consents to the use of such signals. Such signals can arise from user behavior, historical usage patterns, readings from the device's sensors, etc. When users do not permit use of such signals or provide limited authorization to use such signals, the techniques may utilize default settings, aggregated settings, settings based on previously learned parameters, etc.

A user for whom fine-grained permissions settings are determined is presented with one or more options to allow control over the signals that are collected and utilized to determine the fine-grained permissions. For example, the user is provided with an option to provide authorization to control whether the signals include any user information and as which portions of the user information can be collected. In addition, certain user data may be treated in one or more ways before it is stored or used so that personally identifiable information is removed. As one example, a user's identity may be treated so that no personally identifiable information can be determined. As another example, a user's geographic location may be generalized to a larger region so that the user's particular location cannot be determined.

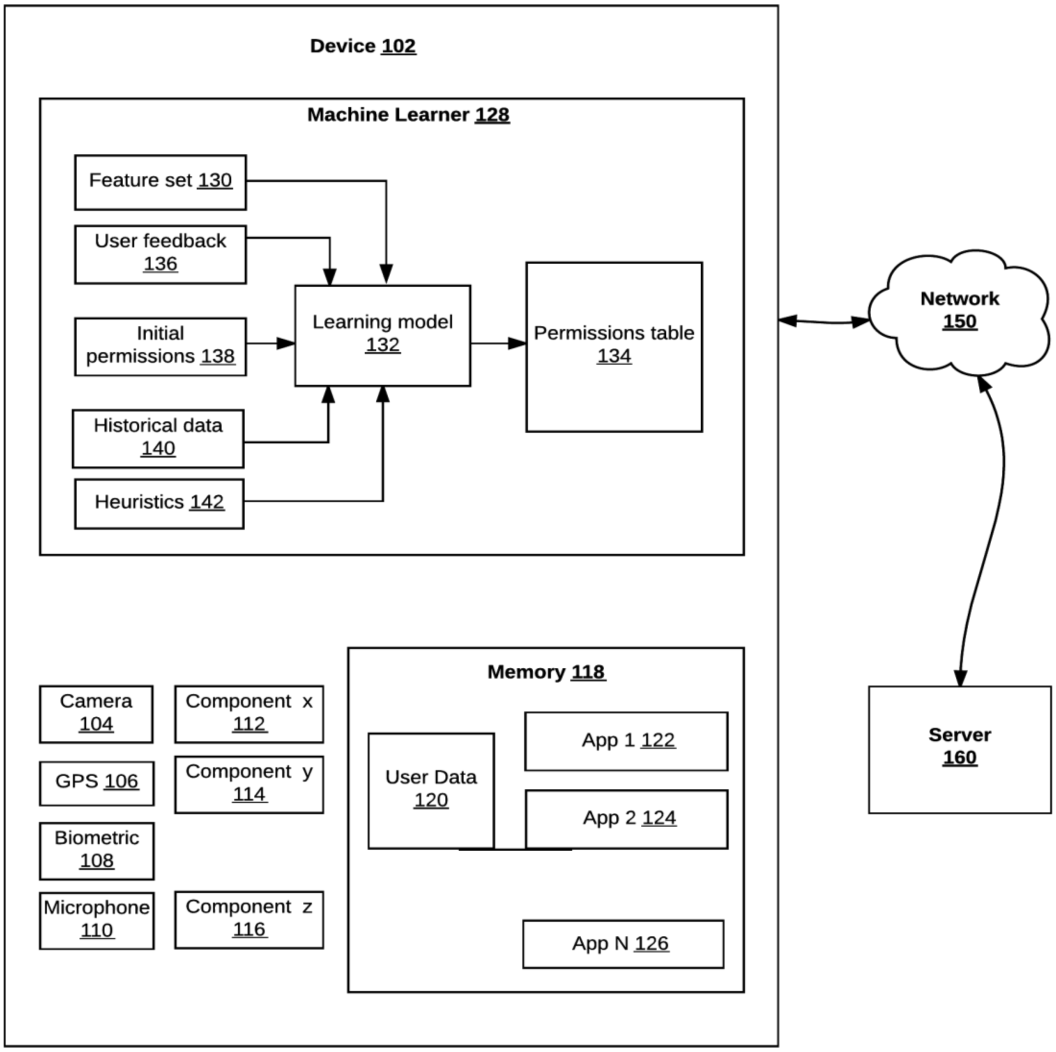


Fig. 1: Automated Setting of Permissions using Machine Learning

Fig. 1 shows an example device that can learn and set permissions for apps. Device (102) includes several hardware components (104-116). Device 102 can be any type of computing device, such as a mobile phone, a tablet, a personal computer, a smartwatch, a wearable computing device, head mounted display, etc. Device 102 includes camera (104), GPS (106), biometric sensors (108), and microphone (110). The techniques described herein can be used to set permissions for any component within the device, including those identified generically in Fig. 1 as component “x” (112), component “y” (114), and component “z” (116).

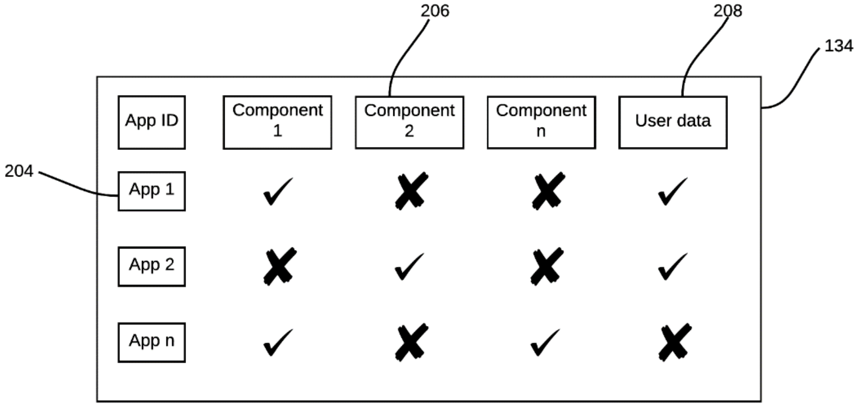
Device 102 also includes a memory (118) that stores user data (120) and one or more apps, labeled “App 1” (122), “App 2” (124), and “App N” (126). In various examples, the user data includes contact lists, photographs, notes, or messages, etc. When the user provides consent to use of the user data, the user data can be used to infer and set permissions for the apps.

Device 102 also includes machine learner (128) which may be implemented as software, hardware, or a combination of software and hardware. The machine learner provides predictions of app permissions, based on available signals. When the user consents to use of user data, the machine learner accesses the user data to determine a variety of signals, e.g., that arise from user behavior. These are collectively referred to as feature set (130). The feature set may include signals from location, time-of-day, app-name, app-identifier, app-type, other permissions requested, time since installation of an app, time since last use of an app, permissions granted by the user in the past, types or levels of permissions (e.g., all contacts, or all contacts in a group, a subset of all known information for one or more persons), time since screen unlock, movement of phone, measurement from sensors, etc. User feedback (136) is also stored and utilized by the machine learner. The user feedback can include whether the user marks a set of permissions recommended by the machine learner as good or bad, modifies the permissions, sets up permission rules, etc. Further, the user feedback can include user-defined rules, e.g., never grant any app access to all contacts.

The machine learner uses the feature set to update a learning model (132). The learning model can utilize any suitable technology such as neural networks, decision trees, active learning, etc. The learning model may be implemented locally on a device on which an app executes, remotely at a server (e.g., server 160) accessible over a network (e.g., network 150), or as a combination. The learning model sets values in a permission table (134). The permission table indicates the permissions granted to respective apps.

At an initial stage, e.g., when the device is being set up, the learning model can utilize an initial set of permissions (138). The initial set of permissions could be one of several possible choices that each specify different levels of access to user data or components. For example, a set of permissions may be denoted, in ascending order of access, as “tinfoil hat”, “paranoid”, “extremely careful”, “privacy sensitive”, and “don’t care”. A user can select one of the permission sets, and the permission table can be initially populated based upon user selection. Alternatively, the permission table can be initialized with the most restrictive initial setting, and updated over time by the machine learner. The machine learner can also determine a confidence level in a predicted permission set. If the confidence level is high, the machine learner can set app permissions automatically, while if the confidence level is low, the machine learner can seek user validation for the predicted permission set. The machine learner presents the selected permission set to the user, e.g., through summary notifications such as “granted access to 72 contacts to social app1”, “denied access to pictures to gaming app x”, etc.

The learning model may also rely on historical data (140) and heuristics (142) in order to configure values in the permission table. Historical data may refer to past patterns of permissions granted to various apps by users, when users provide consent to collection and use of such permissions data. The learning model can apply collaborative filtering techniques to determine values for the permission table based on the historical data, e.g., data about other users' selected permissions.



App ID	Component 1	Component 2	Component n	User data
App 1	✓	✘	✘	✓
App 2	✘	✓	✘	✓
App n	✓	✘	✓	✘

Fig. 2: Permissions Table

Fig. 2 illustrates an example permission table (134) that is maintained by the machine learner. As shown in Fig. 2, each row of the permission table corresponds to an app, e.g., row 204 corresponds to App 1. Each column of the permission table corresponds to a resource for which permissions are configured, such as a hardware component, a software component, user data, etc. While permission to access the user data is represented in a single column in Fig. 2, permissions may be granted at a granular level for individual pieces or groups of user data, e.g., photographs, contacts, etc. A checkmark in the permission table indicates that the corresponding row (i.e., app) has access to the corresponding resource. A cross mark indicates that the app does not have access to the corresponding resource. The entries of the permission table are updated by the learning model, with optional user input.

The permission table can be implemented in a variety of ways. For a given component, e.g., corresponding to a column of the permission table, there may be hierarchical organization that defines permissions at a finer granularity - e.g., location data may be made available at different granularities to different apps, permissions may be stored based on rules based on parameters such as location, time, etc. Further, while Fig. 2 shows permissions as either granted or denied, more elaborate permission sets can be implemented such that more than two levels of permissions can be defined for different apps, e.g., app 1 may be granted no access to photos taken with a camera, app 2 may be granted limited access, e.g., photos taken by activating a device camera from within the app, and app 3 may be granted access to all photos.

Examples of Use

The learning model starts with the initial permissions set and makes entries in the permissions table based upon the feature set, the heuristics, and the historical or archival data. The user assesses the resulting permissions table, or particular entries and provides feedback. If such feedback is available, it is used to train and adapt the learning model. The learning model can generalize patterns over the feature set labeled by means of user feedback. Examples of this sequence of operations are below.

Example 1

During device setup, the user indicated that his initial permission setting is “privacy sensitive,” which corresponds to denial of access to the speaker. Next, the user installs an app that requests permission to use the speaker. Based on the user's initial permission setting, the app is at first denied access to the speaker. Further, historical data indicates that other users typically granted permission to the app to use the speaker. The learning model receives signals that allow it to infer that the user tends to use the speaker at certain locations, e.g., her home, but not at some other locations, e.g., her office. The learning model then updates the permission table so that the app is granted access to the speaker when the user is at home, and the app is denied access to the speaker when the user is at the office.

Example 2

In another example, the learning model generally infers, based on user-permitted signals, that at certain times of the day, such as when the user is typically at the office, access to the speaker is denied. It applies this inferred, time-of-day-based rule, to an app that seeks access to the speaker. The user however tries to use the app at a certain time of day and discovers that the app is not allowed to access the speaker. She can indicate this by explicitly granting permission for the app to the speaker. This user action serves as feedback to the learning model which can update the permission set accordingly.

Example 3

A social app requests permission to access photographs taken using a device camera. The learning model receives signals and infers that photographs taken at a certain location and time, e.g., a high-school reunion party, have been previously shared with a certain contact identified as user's schoolmate. The learning model then allows the social app access to all such photographs, i.e., photographs taken at the high-school reunion party, for the purpose of sharing with any user contact who can be identified as user's schoolmate.

Example 4

During device set-up, the user indicated that her initial permission setting is “extremely careful,” which corresponds to permission to access the internet being generally denied to apps. Next, the user installs a gaming app that requests permission to use the internet. Based on the user’s initial permission setting, the app is at first denied such access. Historical data indicates that most users use the app during certain hours, e.g., non-business hours. At a certain non-business hour, the user explicitly permits the app to access the internet. The learning model then updates the permission table so that the app is granted access to the internet during non-business hours.

CONCLUSION

The techniques of this disclosure enable automated and fine-grained setting of permissions for apps to access resources of a user’s device. Signals from the user’s behavior, user’s feedback and environmental cues are used, with user permission, by a machine learner to set permissions for apps. Heuristics, and historical or archival data are used during the learning process. Fine-grained permission setting is accomplished with minimal user intervention.