April 2020

App Clustering to Identify Anomalous Data Access Permissions

Anonymous

App Clustering to Identify Anomalous Data Access Permissions

ABSTRACT

This disclosure describes the use of clustering techniques to group similar apps on a platform and identify anomalous permissions. Using the described techniques, platform providers can ensure that apps only obtain access to user data that is essential to provide their functionality. Further, platform providers can modularize their API, e.g., break down permissions or features that apps can access, to limit app access to user data. Further, on platforms that conduct human review of apps, app clustering results can be used to provide reviewers with information regarding how the data accesses by a particular app compare with other similar apps. The clustering results can also be utilized to provide app developers suggestions for use of the platform API and data access permissions.

KEYWORDS

- App permission
- App clustering
- App graph
- API access
- User data
- Data access permission
- Improper data access
- Collaboration platform

BACKGROUND

Users make use of software applications such as apps on a social media platform or messaging platform or collaboration platform for various purposes such as playing games,

sharing information, interacting with friends, etc. To provide the functionality, many apps obtain user permission to access user data via the platform. The platform provider provides an application programming interface (API) that apps can use to access user data. However, it can be the case that an app requests and obtains permission to access more user data than is needed to provide the app functionality.

DESCRIPTION

This disclosure describes the use of clustering techniques to group similar apps on a platform and identify anomalous permissions. Using the described techniques, platform providers can ensure that apps only obtain access to user data that is essential to provide their functionality. Further, platform providers can modularize their API, e.g., break down permissions or features that apps can access, to limit app access to user data. Further, on platforms that conduct human review of apps, app clustering results can be used to provide reviewers with information regarding how the data accesses by a particular app compare with other similar apps. The clustering results can also be utilized to provide app developers suggestions for use of the platform API and data access permissions.

Generation of app graph

Logs of data access from an app on a platform are used to generate a graph. Nodes of the graph are apps and edges between nodes represent similarities between two apps. Apps that make accesses to similar data fields via a platform provider API have an edge between them in the input graph, while those that do not access similar data fields or only access a small number of data fields in common do not have an edge between them. For example, a threshold percentage of data fields can be used to determine whether an edge exists between two apps. Fig. 1 illustrates an example graph.

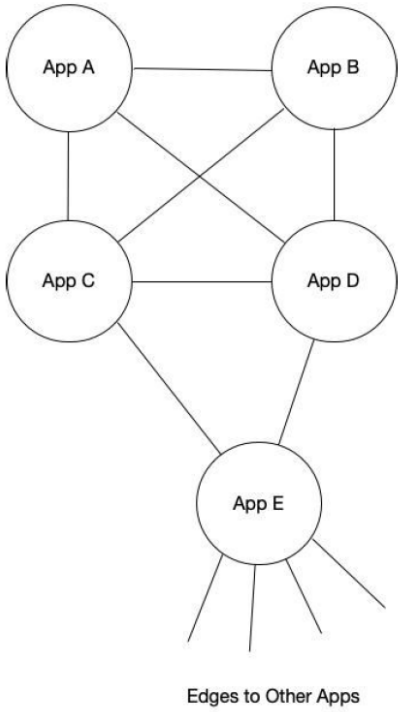

Fig. 1: Example graph

App clustering

The graph thus obtained is provided to a clustering algorithm that identifies clusters of nodes of the graph that are grouped together. Each cluster can include any number of apps. In the example of Fig. 1, apps A, B, C, and D, that have edges between each other are part of the same cluster, while app E is part of a different cluster.

Querying app clusters

For each cluster, queries can be performed to identify specific types of data accesses performed by apps in that cluster. Typically, each cluster has a small set of data fields that are accessed by a high proportion of apps in the cluster and a long tail of data fields that very few apps access. The small set of fields represents the data fields essential to the functionality

represented by the cluster. Data fields in the long tail can either be customizations on top of a common app experience, or can be indicative of improper data access by the particular app.

Once the app clusters are formed, review of permissions for each app can be performed to identify permissions that are rare within the cluster that the app belongs to. Such permissions may be deemed suspicious or reviewable (e.g., by a human reviewer). Further, during review of an app, a list of similar apps to the one being reviewed can be provided to a reviewer, based on the app clusters. Permission settings can also be suggested to the app developer based on API calls made by the app in developer mode.

The app clusters can also be utilized to further modularize the API offered by the platforms. For example, permissions or features of data fields that map to distinct app clusters can be identified. Such permissions or features are indicative of likely distinct uses of the API. Such data can be used to make decisions regarding breaking down such permissions or features.

CONCLUSION

This disclosure describes the use of clustering techniques to group similar apps on a platform and identify anomalous permissions. Using the described techniques, platform providers can ensure that apps only obtain access to user data that is essential to provide their functionality. Further, platform providers can modularize their API, e.g., break down permissions or features that apps can access, to limit app access to user data. Further, on platforms that conduct human review of apps, app clustering results can be used to provide reviewers with information regarding how the data accesses by a particular app compare with other similar apps. The clustering results can also be utilized to provide app developers suggestions for use of the platform API and data access permissions.