Demo: System-E: Enhancing Privacy on Mobile Systems through Content-Based Classification and Storage

Sharath Chandrashekhara, Taeyeon Ki, Karthik Dantu, Steven Y. Ko
Department of Computer Science and Engineering, University at Buffalo
{sc296, tki, kdantu, stevko}@buffalo.edu

ABSTRACT

Mobile systems face privacy challenges including coarse-grained privacy control and the inability to distinguish private and public files. We propose *System-E*, a novel system which can enhance the user privacy on mobile systems (e.g., Android) by (1) enabling users to set finer grained permissions for apps accessing data, and (2) enabling automatic classification of data (e.g., photos) at the storage layer (e.g., by using deep learning) to prevent potentially sensitive data from being stored/accessed with open permissions.

1. OVERVIEW AND MOTIVATION

Popular mobile platforms, such as Android, use an open app distribution model where any app can be installed by the user. These apps, although developed independently, often work together to create a good user experience. Part of this is shared storage, allowing files created by one app to be used by another. For instance, a user who uses a particular camera app to capture photos and use a second app for editing and sharing them. Such a cross-app data sharing model comes at the cost of privacy, which has become a key concern in the mobile ecosystem today [3, 5].

Android uses a coarse-grained permission mechanism, where each app can store data in its own private space or in a shared space, accessible by all apps. In this model, an app can easily get access to data a user might not expect it to access [4]. For instance, when a user grants storage permissions to an app that takes pictures of receipts and allows friends to split the bill, it can also get access to all of the users photos and documents received over a chat application, some of which could be sensitive.

To solve this problem, we envision a system which can ensure sensitive data (e.g., private photos) is made accessible to only apps that have permissions to access them. In such a system a user should be allowed to (1) override the platform's default permissions by identifying trusted and untrusted apps, (2) customize the way in which sensitive data is identified. To this end, we build *System-E*, a filesystem built using the BlueMountain [1] framework.

2. DEMO

Our demo will show a camera app built with *System-E* on Android, which uses a pre-trained model, based on Google

Figure 1: *System-E* Overview

Tensorflow [2] using the Mobilenets architecture to classify photos into public (e.g., landscapes) or private (e.g., people). The classification is done online by intercepting the storage calls of the camera app. Photos identified as private are redirected to the app's private space, while the public photos are stored in the default space. We show that this live classification can be done at the storage layer without any noticeable delay. Our system also allows the user to install any classification model allowing for customization of the types of photos she might consider private.

3. FUTURE WORK

We believe there is a need for a system to address the privacy concerns of mobile system users in a practical way. *System-E* shows that we can use deep-learning and image processing techniques to identify potentially sensitive data and prevent their use in untrusted apps. Our current effort is towards expanding the identification of sensitive data and developing a privacy manager in the app-space which allows the users to control the data accessed by the apps running on their mobile devices.

4. REFERENCES

- [1] S. Chandrashekhara et al. Bluemountain: An architecture for customized data management on mobile systems. MobiCom '17. ACM.
- [2] Google. Tensor Flow for Android, Jan 2018.
- [3] J. Jeon et al. Dr. android and mr. hide: Fine-grained permissions in android applications. SPSM '12.
- [4] A. Srivastava et al. Camforensics: Understanding visual privacy leaks in the wild. SenSys '17.
- [5] Y. Xu et al. Maxoid: Transparently confining mobile applications with custom views of state. In EuroSys '15.