Smartphone Users Want to Be Mocked

Gino Buzzelli, Nick DiRienzo, and Geoffrey Challen
Department of Computer Science and Engineering
University at Buffalo
{ginobuzz,nvdirien,challen}@buffalo.edu

To secure users' sensitive data, Android employs a "take-it-or-leave-it" permissions model. Unfortunately, this model has many well-documented flaws: applications request unnecessary permissions [1, 2], while users don't understand the implications of permissions requested [3]. An alternative "take-it-or-break-it" approach [5] allows users to selectively enable particular permissions, but many applications require access to specific sensitive information to function properly.

Instead of limiting access to data, we propose improving privacy measures by allowing users to provide synthetic or "mocked" data to manipulate data-driven analytics collected by applications. We introduce a system named PocketMocker that enables objective-based context mocking, allowing users to craft their digital identities based on their personal objectives. PocketMocker mocks context by recording and replaying all information that could allow applications to pierce the mocking environment, such as: location, network characteristics, sensor data and user interaction.

Many people are interested in maintaining an attractive online presence on social networks, whether that be with friends or with complete strangers. A fitness application, for example, can determine a health profile for a user based on geolocation and other sensor data. Take, for instance, one of their users, Bob, wants to appear more active, so he "started" a walking regimen. However, instead of physically running around the park, he uses PocketMocker to schedule a mile run every day, which provides mock geolocation and accelerometer data to the mobile fitness application. To his friends, he appears more active, but in reality, he has not changed his habits.

PocketMocker is driven by user interaction, but also needs to initiate, record, and replay mocking sessions. The user can record sensor data on a per-objective basis through the system's user-level application, but the record-and-replay functionality requires low-level support to record the user data to be later mocked and to prevent third-party applications from piercing the mocking environment. We are implementing the system as a prototype by modifying the Android API. In the future, we plan to strengthen the security of the mocking environment by modifying the Linux kernel, as shown in Figure 1, which will allow PocketMocker to mislead applications that bypass the Android Java platform.

PocketMocker consists of three stages of development. In its initial stage, *record and replay*, users will be able to record sensor data on a per-objective basis through a user application. In PocketMocker's second stage, we will explore automated replays by learning from user behavior. In the third and final stage, we will be able to synthesize mocking sessions not previously defined by the user; this would allow PocketMocker to have Bob run on different trails he has not yet explored in his workout routine, for example.

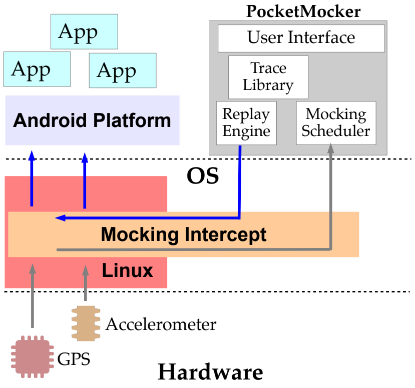

Figure 1: PocketMocker Architecture

PocketMocker is in its early stages of development, and once prototyped, it will be tested on the PhoneLab [4] smartphone testbed. If successful, we believe that PocketMocker can help users retain control over their data while increasing the value of truthful data to companies and advertisers.

REFERENCES

[1] Enck, W., Gilbert, P., Chun, B.-G., Cox, L. P., Jung, J., McDaniel, P., and Sheth, A. N. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. In Proceedings of the 9th USENIX conference on Operating systems design and implementation (Berkeley, CA, USA, 2010), OSDI'10, USENIX Association, pp. 1–6.

[2] Felt, A. P., Chin, E., Hanna, S., Song, D., and Wagner, D. Android permissions demystified. In Proceedings of the 18th ACM conference on Computer and communications security (New York, NY, USA, 2011), CCS '11, ACM, pp. 627–638.

[3] Felt, A. P., Ha, E., Egelman, S., Haney, A., Chin, E., and Wagner, D. Android permissions: user attention, comprehension, and behavior. In Proceedings of the Eighth Symposium on Usable Privacy and Security (New York, NY, USA, 2012), SOUPS '12, ACM, pp. 3:1–3:14.

[4] Nandugudi, A., Maiti, A., Ki, T., Bulut, F., Demirbas, M., Kosar, T., Qiao, C., Ko, S. Y., and Challen, G. Phonelab: A large programmable smartphone testbed. In 1st International Workshop on Sensing and Big Data Mining (SenseMine 2013) (November 2013).

[5] Nauman, M., Khan, S., and Zhang, X. Apex: extending android permission model and enforcement with user-defined runtime constraints. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (New York, NY, USA, 2010), ASIACCS '10, ACM, pp. 328–332.