Poster: Securing Device Inputs for Smartphones Using Hypervisor Based Approach

Xin Zhang, Yongshu Bai, Pengzhan Hao, and Yifan Zhang
Department of Computer Science
SUNY Binghamton
Binghamton, NY
{xzhang99, ybai4, phao3, zhangy}@binghamton.edu

Introduction. Smartphone device inputs, such as inputs from touchscreen, sensors, and GPS, carry sensitive user information, but are vulnerable to passive and active attacks. In one category of the attacks, attackers can passively infer or actively steal sensitive user info from smartphone device inputs. In another category, attackers can tamper with or forge smartphone device inputs to disrupt services relying on those input data or even gain control of the smartphone. We present our ongoing design and implementation of `SDIF`, a Secure Device Input Framework for smartphones.

System architecture. Figure 1 shows the architecture of `SDIF` on an Android smartphone. The core components of `SDIF` are *a small and dedicated bare-metal hypervisor built using ARM hardware virtualization support* and *a user-space sandbox framework*, which collectively ensure `SDIF`'s support for unmodified OSes and apps. `SDIF` secures smartphone device inputs with two novel designs:

• First, `SDIF` enables *a secure path between smartphone input devices and the protected applications*, allowing device input streams to be securely transferred between the two endpoints. On the device endpoint, `SDIF` relies on the dedicated bare-metal hypervisor to achieve secure device I/O management. By trapping and inspecting sensitive activities, the `SDIF` hypervisor can monitor the I/O operations in the system without modifying any code of the OS. On the application endpoint, `SDIF` relies on the user-space sandbox framework to provide necessary utilities for the applications to communicate with the input devices via the secure path. To connect the two endpoints, a communication protocol specifying how device inputs are transmitted between the endpoints via the secure path is needed. We are currently exploring three different communication protocol designs on two different secure paths, aiming to ensure that no attacker can infer or steal user information from device inputs when the user is interacting with the protected applications.

• Second, to ensure that the input devices are properly configured when device drivers read data from the devices, and that device drivers themselves will not tampered with the data before handing them over to the OS, `SDIF` ports the device driver functionalities to the hypervisor, which provides a trusted execution environment. However, this brings a notable challenge, which is the need of minimizing the hypervisor's code size (and hence minimizing its TCB). `SDIF` addresses this challenge by porting only the driver functionalities that are critical for trusted device data reading, which is a small part in driver's code, into the hypervisor, such that the original unmodified device driver and the hypervisor can collectively achieve trusted device data reading. The above two designs collectively guarantee that the device inputs that a protected application receives have not been tampered with.

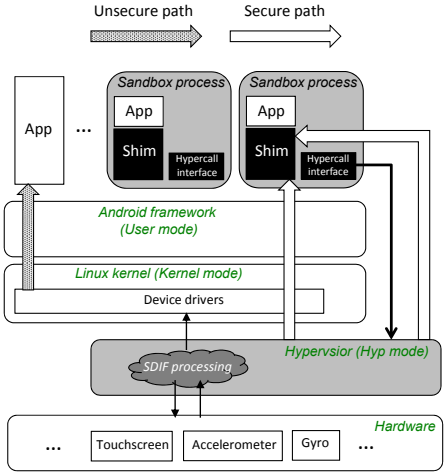

Figure 1: `SDIF` overview (shaded parts are the components of `SDIF`).

Ongoing implementation. We are in the active process of implementing `SDIF` system on real hardware, and plan to validate the our designs by performing real-world experiments on the prototype system. Due to the easiness of flashing and running customized bootloaders, we use ARM development boards featuring commercial smartphone SoCs as our prototype and experiment platforms. Our initial platform consists of an ODROID XU3 development board, a 9 inch 10-points capacitive multi touchscreen, and a WiFi module. The ODROID XU3 features Samsung Exynos 5422 SoC, which is currently used on many popular smartphones such as the Samsung Galaxy S5. We also have a plan to port the implementation to and validate it on ARMv8 architecture.

Acknowledgments. This work is being supported in part by NSF Award #1566375.

MobiSys'17 June 19-23, 2017, Niagara Falls, NY, USA

ACM ISBN 978-1-4503-4928-4/17/06.
DOI: http://dx.doi.org/10.1145/3081333.3089313