Systematic Security and Privacy Auditing of the Android Ecosystem via Automated System-Level Re-Hosting

Thomas Sutter[1,2], Supervision: Timo Kehrer[1], Marc Rennhard[2], Bernhard Tellenbach[3]

University of Bern[1], Zurich University of Applied Sciences[2], Armasuisse W+T[3]

PROBLEM STATEMENT

OS FRAGMENTATION

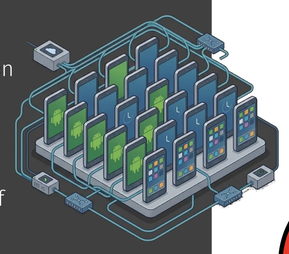
The Android Open Source Project is sequentially modified by silicon vendors, Original Equipment Manufacturers, telecommunication carriers, and third-party app developers before reaching consumers. At each distribution stage, these stakeholders inject proprietary configurations, third-party services, and pre-installed applications directly into read-only system partitions, resulting in a highly diverse landscape of manufacturer-specific firmware variants.

OEM-LOCK & ROOTING



OEM-locking is a restrictive security mechanism enforced by manufacturers via vendor-locked bootloaders to ensure that only officially approved firmware executes on the physical device. Rooting is the process of bypassing these hardware-bound restrictions to gain privileged administrative access, which is often necessary to analyze, modify, or remove applications deeply embedded within read-only system partitions.

REPRODUCIBILITY



Traditional dynamic security auditing often relies on physical devices to interact directly with real hardware and vendor-specific software stacks. However, this hardware-bound approach fundamentally fails to scale across heterogeneous firmware variants due to the prohibitive burdens of device acquisition, continuous maintenance, and complex state management.

THESIS POSTULATE

By automating Android re-hosting to virtual devices, we decouple the application layer from physical devices and emulate the missing peripherals.

This allows us to run complex Android vendor variants dynamically in a fully controlled environment, enabling automated, high-fidelity security and privacy audits at scale.

CORE HYPOTHESIS

By eliminating the reliance on costly and difficult-to-maintain physical device farms, this approach allows for automated, high-fidelity security and privacy auditing of Android's application layer at scale.

Ultimately, the research postulates that this hardware-independent methodology will empower researchers and practitioners to systematically track vulnerability patterns, uncover silent data exfiltration, and expose systemic supply-chain privacy risks embedded deep within the Android ecosystem.

INTRODUCTION

The Android ecosystem's distribution model is inherently multi-tiered and decentralized, traversing from the Android Open Source Project (AOSP) through silicon vendors, OEMs, and telecommunication carriers. At each upstream hop, actors inject proprietary modifications and pre-installed applications, breeding a massive visibility gap that introduces severe flaws like silent privilege escalation and undocumented data collection. While static analysis scales efficiently, it suffers from high false-positive rates and fails to capture complex runtime interactions. Conversely, dynamic auditing provides high-fidelity insights but is severely bottlenecked by the reliance on costly physical device farms and the inability of generic stock emulators to execute hardware-dependent vendor components. To bridge this gap, this research proposes a systematic, automated framework that decouples Android's application layer from physical hardware constraints via re-hosting, enabling the mobile security community to dynamically execute and audit complex downstream Android variants at scale.



OVERVIEW

1

STATIC ANALYSIS AT SCALE

FirmwareDroid (FMD) is an open-source framework that automates the extraction and static analysis of pre-installed software from raw Android firmware images. In its foundational study, FMD evaluated 5,728 commercial firmware samples across multiple vendors, extracting 75,141 unique pre-installed applications. The analysis exposed substantial supply-chain privacy risks, demonstrating the need for systematic studies on scale.

PAPER LINK

2

A GAP IN THE LITERATURE

We conducted a systematic literature review (SLR) to establish a comprehensive taxonomy and classification scheme for current security research. Our findings reveal that instrumentation and network monitoring tools dominate the field, with research primarily targeting app security, privacy, and malware detection. However, the review also identifies critical recurring constraints in automated testing, such as limited code coverage, inadequate app selection, and non-deterministic execution behaviour.

PAPER LINK

3

RE-HOSTING AND EMULATION

To overcome the scalability limitations of hardware-bound testing, we introduce a systematic approach to re-host Android's application layer into a fully emulated environment. The framework extracts proprietary vendor components—including framework files, native libraries, and pre-installed applications—from real-world firmware and integrates them directly into the Android Open Source Project build pipeline to create a device image that can be executed on the Android emulator.

PAPER LINK

4

DYNAMIC ANALYSIS AT SCALE

We conducted a large-scale dynamic analysis of pre-installed applications (PIAs) to measure their runtime network behaviour and hidden privacy risks. Using a highly parallelized re-hosting pipeline that migrated 2,043 vendor-specific PIAs from 53 real-world Android images into virtualized execution environments. By systematically exercising these applications with an ensemble of UI interaction engines and capturing their network traffic, the study exposed a widespread ecosystem of third-party tracking components embedded within privileged system packages.

5

CURRENT WORK

Building on the insights of our dynamic telemetry studies, future research will refine the auditing pipeline by migrating the re-hosting architecture to Cuttlefish, enabling higher-fidelity hardware abstraction and broader compatibility for complex vendor images.

One open question remains: Do emulated systems behave differently from a security perspective?

UNIVERSITY OF BERN

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

armasuisse

CYD | CYBER DEFENCE CAMPUS



Contact Details:
Thomas Sutter
thomas.sutter@unibe.ch
https://7homassutter.github.io/