

# Network-Level Insights into Google-Free Android Operating Systems

Bruno Mirčevski<sup>1</sup>[0009–0007–1008–6708] and Second Author<sup>1</sup>[1111–2222–3333–4444]

Faculty of Computer Science, Bialystok University of Technology, Wiejska 45a,  
15-351, Bialystok, Poland

**Abstract.** This paper presents a comparative analysis of privacy in Android operating systems through network traffic inspection. Using a controlled man-in-the-middle setup and traffic decryption techniques, six Android distributions were evaluated: stock Google Android, GrapheneOS, iodéOS and LineageOS (in 3 variants). The study examined out-bound connections, transmitted identifiers, and network traffic patterns to assess the impact of integrated Google services on user privacy. Results indicate that stock Android maintains the most communication with Google servers, transmitting identifiers and metadata even when idle. In contrast, privacy-focused systems minimize unsolicited traffic and offer stronger user control. The findings demonstrate that alternative Android distributions can substantially improve privacy without sacrificing core functionality, highlighting the potential of open-source ecosystems as viable, privacy-respecting alternatives to Google-dependent mobile environments.

**Keywords:** Android · Privacy · Network traffic.

## 1 Introduction

Android is the dominant mobile operating system and acts as a primary gateway to digital world for billions of users. It's crucial for core OS to be a reliable, stable and trustworthy platform, that allows users to run apps and services of their choice. While mobile app privacy has been widely studied [2, 8, 6, 11, 10], less attention is typically paid to the operating system layer and to privileged service frameworks that silently enable additional functionality. Standard unprivileged apps are constrained by Android's permission model, although some attempt to circumvent it in creative ways [7, 5]. In contrast, privileged system apps operate with higher access to the device and OS, which makes them harder to understand and control. Most user-facing privacy controls focus on applications and permissions, yet a substantial amount of data exchange can occur below the app layer, through OS services, vendor components, and Google service frameworks. This work therefore compares the network behavior of multiple Android distributions and their preinstalled/privileged system components under controlled, reproducible conditions.

### 1.1 Android dependency on privacy-threatening services

While Android’s core is the Android Open Source Project (AOSP), most consumer devices ship with additional proprietary components layered on top. They are often implemented as privileged system apps and service frameworks to provide baseline functionality such as push notifications, location assistance, app distribution/updates, device integrity checks and many more. Users and app developers commonly assume their presence and availability by default.

These components exist for a practical reason. Centralized services can significantly improve usability and efficiency, for example by reducing battery drain compared to per-app background connections, and by providing faster, more accurate location through fused network and sensor-based methods. However, this design also tightly couples many functions into a single service stack, so enabling useful capabilities may implicitly enable others that a user would not choose, such as advertising identifiers, telemetry, and behavioral tracking. In practice, the result is a large proprietary bundle that is difficult to audit and decompose into only the features a user actually wants.

On Android, these baseline capabilities are most commonly delivered through Google’s service stack. Because Google’s business model is strongly advertising-driven, many users may be unwilling to entrust this provider with broad access to device and their data. In principle, users should be able to choose which services they rely on and have confidence that these choices are enforced by the platform’s permission and isolation mechanisms. This demand for transparency and controllability motivates alternative approaches such as microG and GrapheneOS’ sandboxed Google Play. These solutions aim to preserve device functionality and app compatibility while reducing privilege and improving user control over Google-related components.

### 1.2 Privacy-focused Android operating systems

Privacy-focused Android operating systems (also referred to as *custom ROMs*) are distributions derived from AOSP that aim to reduce data exposure at the OS level by limiting preinstalled privileged components, tightening default settings, and providing more transparent controls over permissions and network access. In contrast to many factory builds that bundle large proprietary service stacks, privacy-oriented systems typically try to (i) minimize default background communication, (ii) reduce the use of persistent identifiers, and (iii) give the user stronger enforcement mechanisms to disable or constrain components that are not strictly required for everyday use.

In this work we focus on three representative approaches. GrapheneOS prioritizes a hardened security model and strong isolation. Google components are not included by default, but can be installed in a dedicated sandboxed mode as ordinary apps, without the special privileges that Google Mobile Services (GMS) typically hold on factory systems. This design targets improved containment and user control while preserving high compatibility when the user decides to install Google services.

LineageOS provides a lightweight, broadly supported AOSP-based system with minimal preinstalled software and extensive device compatibility. It can be deployed in multiple configurations: without Google services, with official proprietary Google Play Services and Play Store, or with microG as an open-source reimplementation of key Google APIs. These configurations represent different trade-offs between privacy, compatibility, and reliance on proprietary services.

Finally, iodéOS is a privacy-oriented distribution based on LineageOS that combines a de-Googled baseline with additional privacy features and preconfigured app ecosystem choices, shipping with microG. For completeness, CalyxOS follows a similar privacy-oriented direction (AOSP-based, microG-enabled, and designed for practical daily use). It was considered for the study, but was excluded because it was not receiving updates during the measurement period.

### 1.3 Challenges of network traffic analysis on Android

Network traffic analysis on Android is difficult by design and that is largely a security benefit. Modern protocols increasingly hide data behind encryption (TLS 1.3, QUIC/HTTP3, encrypted DNS). Even when traffic is captured, compression and binary encodings can make payloads hard to interpret without schemas or additional context. As a result, MITM-based decryption is inherently incomplete. Certificate pinning and anti-tamper mechanisms can block interception or break app functionality. Circumvention of these security mechanisms do not reliably yield a full plaintext view. While prior work demonstrates these challenges in a reproducible emulator-based setup [1], this work uses a physical device to capture behavior under more realistic firmware, hardware, and service conditions. We accept that some flows will remain opaque, not all network communication will be revealed or understood. Our focus is on differences between studied Android systems and on general patterns.

## 2 Related work

Prior work has shown that privacy risks on Android extend beyond user-installed applications and can arise from the operating system layer and preinstalled components. Measurement studies of vendor-modified Android variants report background communication to OS vendors and third parties even under minimal configuration. Transmitted data includes persistent identifiers, app and device metadata, which creates possibility of re-linking resettable identifiers when they co-occur with long-lived identifiers [4]. Other measurements report that Google Play Services and the Play Store store cookies, tokens, and device identifiers even in the absence of explicit user activity [3]. Broader ecosystem-level investigations further highlight how data collection can be driven not only by first-party apps and platforms, but also by advertising and analytics infrastructure (which is sometimes preinstalled), enabling linkage between nominally pseudonymous identifiers and user accounts [9].

[Alternative OS are not well studied...]

### 3 Experiment design

### 4 Results

### 5 Conclusion

## References

1. Jimenez-Berenguel, A., Campo, C., Moure-Garrido, M., Garcia-Rubio, C., Díaz-Sánchez, D., Almenares, F.: Parrot: Portable android reproducible traffic observation tool (Sep 2025). <https://doi.org/10.48550/arXiv.2509.09537>
2. Jin, H., Liu, M., Dodhia, K., Li, Y., Srivastava, G., Fredrikson, M., Agarwal, Y., Hong, J.I.: Why are they collecting my data? inferring the purposes of network traffic in mobile apps. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2**(4) (Dec 2018). <https://doi.org/10.1145/3287051>, <https://doi.org/10.1145/3287051>
3. Leith, D.J.: Cookies, identifiers and other data that google silently stores on android handsets. Tech. rep., Trinity College Dublin (Feb 2025), [https://www.scss.tcd.ie/doug.leith/pubs/cookies\\_identifiers\\_and\\_other\\_data.pdf](https://www.scss.tcd.ie/doug.leith/pubs/cookies_identifiers_and_other_data.pdf)
4. Liu, H., Patras, P., Leith, D.J.: Android mobile os snooping by samsung, xiaomi, huawei and realme handsets (Oct 2021), [https://www.scss.tcd.ie/doug.leith/Android\\_privacy\\_report.pdf](https://www.scss.tcd.ie/doug.leith/Android_privacy_report.pdf)
5. LocalMess: Disclosure: Covert web-to-app tracking via localhost on android (2025), <https://localmess.github.io/>
6. Razaghpanah, A., Nithyanand, R., Vallina-Rodriguez, N., Sundaresan, S., Allman, M., Kreibich, C., Gill, P.: Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem (01 2018). <https://doi.org/10.14722/ndss.2018.23009>
7. Reardon, J., Feal, Á., Wijesekera, P., On, A.E.B., Vallina-Rodriguez, N., Egelman, S.: 50 ways to leak your data: An exploration of apps' circumvention of the android permissions system. In: 28th USENIX Security Symposium (USENIX Security 19). pp. 603–620. USENIX Association, Santa Clara, CA (Aug 2019), <https://www.usenix.org/conference/usenixsecurity19/presentation/reardon>
8. Ren, J., Lindorfer, M., Dubois, D.J., Rao, A., Choffnes, D.R., Vallina-Rodríguez, N.: Bug fixes, improvements, ... and privacy leaks - a longitudinal study of pii leaks across android app versions. In: Network and Distributed System Security Symposium (2018), <https://api.semanticscholar.org/CorpusID:4231807>
9. Schmidt, D.C.: Google data collection. Tech. rep., Digital Content Next (DCN) (Aug 2018), <https://digitalcontentnext.org/wp-content/uploads/2018/08/DCN-Google-Data-Collection-Paper.pdf>
10. Sutter, T., Tellenbach, B.: Firmwaredroid: Towards automated static analysis of pre-installed android apps (May 2023). <https://doi.org/10.1109/MOBILSoft59058.2023.00009>
11. Wang, Z., Li, Z., Xue, M., Tyson, G.: Exploring the eastern frontier: A first look at mobile app tracking in china. In: Sperotto, A., Dainotti, A., Stiller, B. (eds.) *Passive and Active Measurement (PAM 2020)*. pp. 314–328. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020)