Network-Level Insights into Google-Free Android Operating Systems

Bruno Mirčevski[1][0009−0007−1008−6708] and Second Author[1][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, five Android distributions were evaluated: stock Google Android, GrapheneOS, iodéOS and LineageOS. The study examined outbound connections, transmitted identifiers, and telemetry patterns to assess the impact of integrated Google services on user privacy. Results indicate that stock Android maintains the most background 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 [1, 5, 3, 6], 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 [4, 2]. 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 thesis 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 exists 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 Understanding the threat model

1.3 Privacy-focused Android operating systems

1.4 Challenges of network traffic analysis on Android

2 Related work

3 Experiment design

4 Results

5 Conclusion

References

1. 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

2. LocalMess: Disclosure: Covert web-to-app tracking via localhost on android (2025), https://localmess.github.io/
3. 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
4. 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
5. 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
6. 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)