From Logging to Leakage: A Study of Privacy Leakage in Android App Logs

Zhiyuan Chen
zc9482@g.rit.edu
Department of Software Engineering, Rochester Institute of Technology
Rochester, NY, USA

Soham Sanjay Deo
sd5456@g.rit.edu
School of Information, Rochester Institute of Technology
Rochester, NY, USA

Poorna Chander Reddy Puttaparthi
pp5109@g.rit.edu
Department of Software Engineering, Rochester Institute of Technology
Rochester, NY, USA

Yiming Tang
yxtvse@rit.edu
Department of Software Engineering, Rochester Institute of Technology
Rochester, NY, USA

Xueling Zhang
xueling.zhang@rit.edu
Department of Software Engineering, Rochester Institute of Technology
Rochester, NY, USA

Weiyi Shang
wshang@uwaterloo.ca
Department of Electrical and Computer Engineering, University of Waterloo
Waterloo, ON, Canada

Abstract

Android phones are among the most popular mobile devices today, providing users with a wide array of convenient services through various apps. These apps generate software logs during their runtime, which record their behavior, status, and error information. However, these logs can also inadvertently capture sensitive information and user privacy data, often without the developer's awareness. In this study, we constructed a dataset comprising 67,702 log records from 83 Android apps. Our analysis of this dataset identified 610 instances of privacy leakage, which indicates the prevalence of such issues in Android app logs. Additionally, our analysis identified characteristics of Android app logs with exposed sensitive information and revealed a gap between developers' awareness of privacy protection and privacy leakage in real-world scenarios.

CCS Concepts

• **Security and privacy → Software security engineering**.

Keywords

Software logs, Logging, Android, Privacy

ACM Reference Format:
Zhiyuan Chen, Soham Sanjay Deo, Poorna Chander Reddy Puttaparthi, Yiming Tang, Xueling Zhang, and Weiyi Shang. 2024. From Logging to Leakage: A Study of Privacy Leakage in Android App Logs. In *39th IEEE/ACM International Conference on Automated Software Engineering (ASE '24), October 27-November 1, 2024, Sacramento, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3691620.3695340

ASE '24, October 27-November 1, 2024, Sacramento, CA, USA

ACM ISBN 979-8-4007-1248-7/24/10
https://doi.org/10.1145/3691620.3695340

1 Research problem

Mobile devices have brought people convenient access to information and facilitated their communication. Logs are the files generated by running the apps, mobile software generates software logs during run-time[3] and installation [6] for developers. Software logging is the practice of recording and storing apps' runtime information [2]. This practice is an essential aspect of software development and maintenance, as it provides insights into the app's behavior, performance, run-time status and potential errors. Nevertheless, logs may store information that developers are unaware of, as the logging content can be derived from variables, not just static text, which might result in significant privacy leaks.

The issue of privacy leakage in Android apps has gained significant research attention. For example, PPChecker [10], an approach designed to systematically analyze the privacy policies of Android apps. This tool aims to assist in improving and standardizing app privacy policies. Additionally, many analysis techniques indicated difference ways to detect privacy leakages. For instance, Jain et al. [5] detects privacy leaks between apps by sharing preferences. FlowDroid [1] uses static taint analysis to track information flows and identify potential leaks. Similarly, VetDroid [11] serves as a dynamic analysis platform to detect how apps access users' private and sensitive information.

Although many studies can detect privacy leaks in android app logs, their methods are time-consuming and these studies lack a log perspective to investigate privacy leaks in Android app logs. We aim to study and analyze privacy leakage in Android apps from the perspective of software log analysis. We aim to analyze the characteristics of the leaked information, we also bring some new insight to developers to raise developers' consciousness by analyzing Android developers' awareness of privacy protection.

2 Study overview

This study addresses the issue of privacy leakage in Android apps by analyzing it from three key perspectives: (1) examining Android apps available in the market and analyzing their logs for privacy leaks, (2) identifying the characteristics of these privacy leaking

logs, and (3) assessing real-world developers' awareness of privacy leakage in Android apps.

2.1 Exploring Privacy Leaks

To ensure the breadth and diversity of our analysis, we have chosen the top 100 Android apps from PlayDrone [7], and successfully installed and analyzed 83 of them. Playdrone is the first scalable Google Play Store crawler and uses it to index and analyze over 1,100,000 apps in the Google Play Store every day. To achieve this, we employed the Android Debug Bridge (ADB) [4] to automate the installation of apps. To simulate user interactions, we use Monkey tool which is a program that runs on the device to stress-test applications which are developing, in a random yet repeatable manner, thereby generating comprehensive software logs [9].

Our own Android app log dataset includes users' personal information, device information, and other relevant data. Since this information is known to us, it allows us to efficiently identify private data present in the logs by searching keywords. We also considered selecting weak cryptographic algorithms, such as SHA-1 and MD5, as they are susceptible to being compromised. Once identified, we further categorize and analyze the private information leaked.

2.2 Characteristics of Privacy Leakage in Android App logs

To better understand the nature of privacy leakage, we first focus on analyzing its location in the Android app logs. Android logs are divided into five levels: ERROR, WARN, INFO, DEBUG, and VERBOSE, in descending order of priority. However, in practice, the logs are typically printed with the first three levels-ERROR, WARN, and INFO-and automatically suppressed with VERBOSE and DEBUG. Studying logging levels involved in privacy leakage is essential. Since lower logging levels may not result in log printing, privacy leakage at lower logging levels may pose a lower risk, and vice versa.

2.3 Awareness of developers

Numerous studies have highlighted that while developers possess some degree of awareness regarding privacy leakage issues, this awareness varies significantly across individuals and contexts[8]. We conduct a study on online developer Q&A forums, such as Stack Overflow, for posts related to privacy leakage in Android app logs. By analyzing these posts, we aim to understand developers' awareness of privacy protection in real-world Android app logs, and compare their practices with the privacy leakage instances we have identified in our log analysis.

3 Results achieved so far

From our 83 successfully installed apps, we generated a total of 667,702 log records, from which we identified 610 instances of privacy leakage. Our analysis revealed that the majority of these leaks occurred within the DEBUG and INFO log levels, which account for a significant portion of the analyzed logs. Furthermore, the awareness of privacy protection among developers remains inadequate, leading to a high proportion of app suspected of privacy leakage due to the low prioritization of privacy concerns in practice. We found 613 posts on developers' Q&A forums , after manual screening we found 59 posts related to software log privacy leaks. In addition, developers' concerns about software log privacy leaks include log usage, log content concerns, and the replacement of sensitive information.

4 Conclusion

In this study, we conducted an empirical study of privacy leakage vulnerabilities in Android app logs. Specifically, we built our own Android app log dataset, detected privacy leakages within this dataset, analyzed the characteristics and patterns of these privacy leakage features, and evaluated the awareness of developers regarding privacy protection. Our findings reveal significant gaps in developers' awareness and practices, which contribute to the potential risks associated with long-term log file storage and distribution.

References

[1] Steven Arzt et al. "Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps". In: *Acm Sigplan Notices* (2014).

[2] Boyuan Chen and Zhen Ming Jiang. "Characterizing and Detecting Anti-Patterns in the Logging Code". In: *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. 2017. DOI: 10.1109/ICSE.2017.15.

[3] Wenhao Fan et al. "DroidInjector: A process injection-based dynamic tracking system for runtime behaviors of Android applications". In: *Computers & Security* 70 (2017), pp. 224–237. DOI: https://doi.org/10.1016/j.cose.2017.06.001.

[4] Google. *Android Debug Bridge (adb)*. URL: https://developer.android.com/tools/adb (visited on 03/20/2024).

[5] Vineeta Jain et al. "SniffDroid: Detection of Inter-App Privacy Leaks in Android". In: *2017 IEEE Trustcom/BigDataSE/ICESS*. 2017. DOI: 10.1109/Trustcom/BigDataSE/ICESS.2017.255.

[6] Jinwoo Lee et al. "Analysis of application installation logs on Android systems". In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. SAC '19. Limassol, Cyprus: Association for Computing Machinery, 2019. DOI: 10.1145/3297280.3297489.

[7] *PlayDrone metadata*. 2018. URL: https://archive.org/details/android_apps&tab=about.

[8] Awanthika R. Senarath and Nalin Asanka Gamagedara Arachchilage. "Understanding user privacy expectations: A software developer's perspective". In: *Telematics and Informatics* (2018). ISSN: 0736-5853. DOI: https://doi.org/10.1016/j.tele.2018.05.012.

[9] *UI/Application Exerciser Monkey*. 2019. URL: https://developer.android.com/studio/test/monkey.html.

[10] Le Yu et al. "Can We Trust the Privacy Policies of Android Apps?" In: *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2016. DOI: 10.1109/DSN.2016.55.

[11] Yuan Zhang et al. "Vetting undesirable behaviors in android apps with permission use analysis". In: CCS '13. Berlin, Germany: Association for Computing Machinery, 2013. DOI: 10.1145/2508859.2516689.