A Study of User Privacy in Android Mobile AR Apps

Xiaoyi Yang
xy3371@rit.edu
Rochester Institute of Technology
Rochester, New York, USA

Xueling Zhang
xueling.zhang@rit.edu
Rochester Institute of Technology
Rochester, New York, USA

ABSTRACT

With the development of augmented reality (AR) technology, the use of mobile AR applications (MAR apps) is rising rapidly in various aspects of people's everyday lives, such as games, shopping, and education. When compared to traditional apps, AR apps typically need access to the smartphone's camera all the time and collect and analyze significantly more data, such as sensor data, geolocation, and biometric information. Due to the sensitivity and volume of data collected by MAR apps, new privacy concerns are raised. In this paper, we describe a preliminary empirical study of Android MAR apps in terms of the sensitive data collected by MAR apps, whether the collected data is well protected, and whether the data practice is publicly available so that users can learn about the data safety and make informed decisions when deciding which apps to install. In this study, we analyzed 390 real-world MAR apps and reported the dangerous permissions they requested, the data leaks detected in them, and the availability of their data safety.

CCS CONCEPTS

• **Software and its engineering**; • **Security and privacy** → *Human and societal aspects of security and privacy*;

KEYWORDS

privacy leak, user privacy, data safety, mobile application

ACM Reference Format:
Xiaoyi Yang and Xueling Zhang. 2022. A Study of User Privacy in Android Mobile AR Apps. In *37th IEEE/ACM International Conference on Automated Software Engineering (ASE '22), October 10–14, 2022, Rochester, MI, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3551349.3560512

1 INTRODUCTION

Augmented Reality (AR) is a technology that incorporates real-world environments and virtual objects and provides users with an interactive experience[11][8]. The success of Pokémon GO in 2016 brought AR to the public attention[28]. As the popularity of AR continuously increased, many developers started to build more AR apps, specifically mobile AR (MAR) apps.

In order to deliver AR-featuring experiences, AR apps collect significant amounts of personal data, including information provided by users (*e.g.*, private information provided by user when registered for the app such as username, password, and email address), information generated by users (*e.g.*, user's physical appearance, gestures, and mannerisms), and information inferred about users (*e.g.*, biometric identification or advertising profiles inferred by manipulating the provided or generated data).

Compared to other smartphone apps, AR apps pose novel issues for user privacy due to the scope, scale, and sensitivity of the information they collect. MAR apps use the camera to get the real-world image, which may expose the user's location without GPS. Additionally, using the camera in public for a long time may disturb bystanders since they will inevitably be filmed or even recognized for computer-generated perceptual information. The user-application interaction may be required to collect personal information displayed in the MAR app. In the case of apps supporting multi-user interactive experiences, one user's personal information could reflect on another user's screen. Furthermore, the MAR app's experience highly relies on the user's motion, such as walking and turning, which depends on the data collected by sensors integrated inside the mobile device. Many pieces of research[5, 6, 24, 25] have suggested that the sensor data can cause significant information leakage on users.

The literature has explored the security and privacy issue of AR apps at the system level [16, 22, 23, 26, 31]. However, little work has focused on the data practice in MAR apps, such as what kind of user data is collected, how the data is used, and whether users are aware of those data practices. Therefore, this paper presents a preliminary empirical study on real-world Android MAR apps to understand their private data practice. Specifically, we try to answer the following research questions:

- **RQ1**: hat personal information is collected by MAR apps?
- **RQ2**: How do MAR app companies use and secure the information they have collected from users?
- **RQ3**: Do MAR app companies clearly disclose their data practices to users?

We analyzed 390 real-world MAR apps to answer these research questions. We started our study by collecting MAR apps' data set and downloading the most up-to-date APK files for analysis. We then used existing static analysis techniques to collect the permission requested by those MAR apps and analyze data flow of sensitive data inside those apps. We also investigate the Data Safety information they provided to user, which is required by Google Play Store for each app.

We have the following major findings[1]:

ASE '22, October 10–14, 2022, Rochester, MI, USA

ACM ISBN 978-1-4503-9475-8/22/10…$15.00
https://doi.org/10.1145/3551349.3560512

[1]The data set and results of this study will be made available upon acceptance not to violate double-blind rules

- We extracted permissions requested by the collected MAR apps and found that the most frequently requested dangerous permission are CAMERA, EXTERNAL_STORAGE, FINE_LOCATION, PHONE_STATE.
- We performed static taint analysis on the 390 MAR apps and identified 30 data leaks from 9 apps. All the leaked sources are sensor data, includes *accelerometer*, *gyroscope*, *magnetic field*, *step counter*, *light sensor*, *gravity sensor*, *pressure*, and *rotation vector*.
- We investigated the Google Play Store for each app under analysis and found that 69.49% of them have no Data Safety information. Among the 119 apps with Data Safety, 34 show "Data isn't encrypted" or do not disclose relevant information.

2 BACKGROUND

This section focus on introducing three aspects of this study. Including Android permission, Taint analysis for data leak detection, and Data Safety of Android apps.

2.1 Android permission

Android utilizes the Linux security model and layers through a user-based permission system. The Android permission system aims to control the access of mobile apps to sensitive resources. Through the permission system, an app can access resources such as the device's GPS location, camera, network connections, and other sensors[41]. Each android permission is assigned a protection level such as *normal*, *dangerous*, and *signature*, to indicate how sensitive the information or action the permission is associated with[14]. In this study, we focused on *dangerous* permissions such as CAMERA as they allow access to highly-sensitive information, such as photo album, location, and phone number[13].

2.2 Taint analysis for leak detection

To ensure that users' data is only used in compliance with the relevant privacy policies, it is necessary to analyze how data flows within the application. Taint analysis is a type of information flow analysis in which tainted objects are tracked using data flow analysis. For smartphone apps, a data leak occurs when sensitive sources (phone numbers, device identifiers, contact lists) flow to sinks (Internet, SMS transmission). Taint analysis is often used to detect privacy leaks: it taints sensitive data as its source, propagates the tainted information through the application, and issues a warning if tainted data reaches a sink.

2.3 Data safety of Android apps

In 2021, Google announced that developers must provide information about security and privacy for display in the Data Safety section in their Google Play Store page to increase information transparency. Developers need to notify the users of data collected from their phones, data shared with the third party, and security practices in the app. According to Google's policy, all developers must provide such information by July 20, 2022[15].

3 STUDY DESIGN

This study aims to learn the privacy data practice in real-world MAR apps. In this section, we discuss the methodology we followed to answer each of research questions.

To answer **RQ 1** regarding the personal information collected by MAR apps, we investigate the permissions requested by MAR apps, as permissions directly reflect what personal data may be collected. To answer **RQ 2** about the usage of data collected by MAR apps, we perform static taint analysis on MAR apps to determine if users' sensitive data is improperly processed. Specifically, we collect a list of sensitive data sources and risky sinks, which we then use as input for taint analysis to detect source-to-sink data leakage. To answer **RQ 3** about disclosing data practices to app users, we check the availability of the Data Safety section in the Google Play Store for each MAR app under study.

3.1 Data Set

To construct our data set, we first created a list of 420 MAR apps based on the pre-defined lists from Google under the "Augmented Reality" category, namely "AR APPs" [34] and "AR Games" [35]. For each MAR app, we then download its most recent version of the APK file from Androzoo [2]. Androzoo is a well-known data set of Android apps that is continuously updated from several major application markets, we successfully downloaded the 390 APK files.

To better present our data set, we collected the category information for each app. Among the 390 apps, we found category information for 331 apps in the Google Play Store. For the remaining 59 apps, we collected their category information from AppBrain [1], a company specializing in app marketing and statistics. Figure 1 shows the category distribution of our data set and the corresponding app percentage.

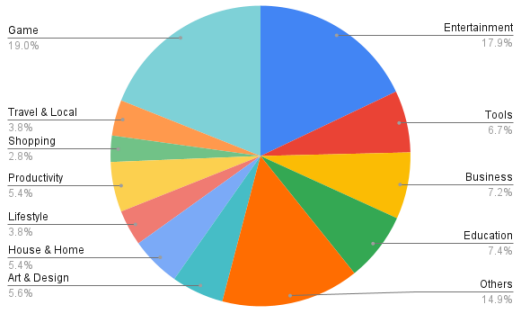


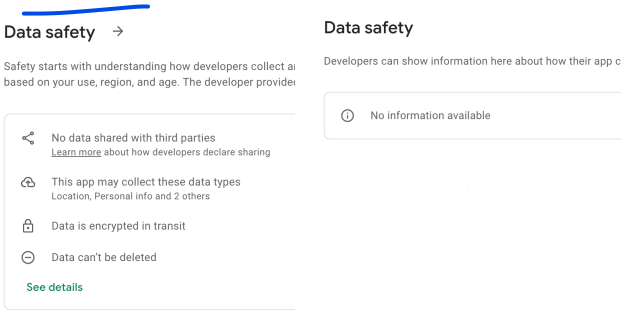
Figure 1: MAR apps distribution in Categories

3.2 Investigation on MAR apps

Based on our research questions, our investigation of MAR apps consists of three parts: permissions extraction, data leak detection, and Data Safety inspection.

3.2.1 Permission extraction. We utilize Androguard [10] to extract permission from each MAR app. Androguard is an open-source python library capable of extracting different kinds of information from the individual components of an APK file. Androguard takes the APK files as input to extract the list of permissions requested by the app under analysis.

3.2.2 Data Leak Detection. We use Flowdroid [4], the state-of-the-art static taint analysis tool for Android apps, to analyze our MAR apps. Flowdroid takes an APK file, a list of sources, and sinks as input and reports a leak if it detects data flow from a source to a sink. SUSI [3] is a machine-learning tool for the automated classification and categorization of Android Sources and Sinks. We construct the list of sources by combing the generic sources from SUSI and the sensor sources from SEEKER [36]. The construed source list includes categories such as *UNIQUE_IDENTIFIER*, *CONTACT*, and *LOCATION_INFORMATION* from SUSI; *MOTION_SENSOR*, *POSITION_SENSOR*, and *ENVIRONMENT_SENSOR* from SEEKER. As sinks, we use the SUSI sink list, including categories such as *LOG*, *NETWORK* and *FILE*.



(a) Available Data Safety **(b) Unavailable Data Safety**

3.2.3 Data Safety Inspection. We manually inspect the availability of the Data safety section for each MAR app. An example of available Data safety section from the app *com.grymala.arplan* is shown in Figure 2a. If an app's Data safety information has not been submitted or have been rejected, "No information available" will be in the Data Safety section. An example of unavailable Data safety section from the app *com.runningpixel.wrldcraft* is shown in Figure 2b.

4 PRELIMINARY RESULTS

In this section, we present the results of our study and answer the research questions.

> **RQ1** *What personal information is collected by MAR apps?*

Among all the permissions extracted from 390 MAR apps, we collected only the dangerous ones. Table 1 shows the most frequently requested dangerous permissions (column 1), the number of apps requesting the permission (column 2), and the percentage of apps within our data set (column 3). The five most frequently requested dangerous permissions are *CAMERA*, *WRITE EXTERNAL STORAGE*, *READ EXTERNAL STORAGE*, *ACCESS FINE LOCATION*, and *READ PHONE STATE*, with more than 25% of MAR apps requested. These permissions allow apps to access sensitive data such as the user's photo album, precise location, and phone number.

> **RQ2** *How do MAR app companies use and secure the information they have collected from users?*

Permission Name	Count	Percentage
CAMERA	352	90.26%
WRITE_EXTERNAL_STORAGE	324	83.08%
READ_EXTERNAL_STORAGE	250	64.10%
ACCESS_FINE_LOCATION	127	32.56%
READ_PHONE_STATE	100	25.64%
ACCESS_COARSE_LOCATION	88	22.56%
RECORD_AUDIO	88	22.56%
GET_ACCOUNTS	42	10.77%
READ_CONTACTS	18	4.62%
CALL_PHONE	12	3.08%
ACTIVITY_RECOGNITION	9	2.31%
WRITE_CONTACTS	6	1.54%
READ_CALENDAR	5	1.28%
WRITE_CALENDAR	5	1.28%
ACCESS_BACKGROUND_LOCATION	3	0.77%

Table 1: Dangerous Permissions requested by MAR apps

Our experiment results show that 30 data leaks were detected from 9 apps. Among the 9 apps, 3 apps have 1 leak, 1 app has 2 leaks, 3 apps have 4 leaks, 1 app has 5 leak, and 1 app has 8 leaks. All leaked sources are sensor data, includes *accelerometer*, *gyroscope*, *magnetic field*, *step counter*, *light sensor*, *gravity sensor*, *pressure*, and *rotation vector*, and all sink data is in the Log category.

> **RQ2** *How do MAR app companies use and secure the information they have collected from users?*

Despite Google requires that all developers update their the Data Safety section in Google Play Store by July 20, 2022, our study demonstrates a delay in that. By the time we checked the Google Play Store for our data set on July 28, 2022, only 119 out of 390 AR apps (30.5%) had the Data Safety information displayed on the app detail page in Google Play Store, while the remaining 271 apps (69.49%) show "No information available." In addition, among the 119 apps with available Data Safety information, 34 apps show "data isn't encrypted" or do not have relevant information provided, indicating that these apps either do not encrypt the data they collect or choose not to report it.

5 DISCUSSION

Regarding the dangerous permissions we have found in the study, users may correspondingly protect privacy by the defense mechanism provided by the Android system. Users can restrict the information that apps access by granting one-time permission while using apps or only giving apps access to specific files that are required. The threat of camera permission is more difficult to control because of the risk of disclosing location[38] or bystander's privacy from the camera display[9]. One possible solution is auto-detecting the bystander and the environment in the display and taking immediate reactions to prevent privacy disclosure. For instance, Steil et al. researched to detect the highly-sensitive situation and disable the camera instantly to preserve the privacy of bystanders[33]. Hasan et al. used convectional neural network to detect bystanders in static images[18].

According to our experiment result, all sinks are in the Log category. The field of sinks in the Log category includes Error Log, Warning Log, Debug Log, and Verbose Log. Information written to log files can be of a sensitive nature and give valuable guidance to an attacker or expose sensitive user information. [37]. While leaking such sensor data may not directly link to malicious behavior, it does expand the attack surface for attackers. For example, the sensor data can be used to infer a user's tap inputs, which could be a user's credential[27, 29]. To mitigate such issues, developers should avoid writing any sensitive data into the log files. The Android platform (Google), which has access to sensor data, should prevent such data from flowing to risky places.

6 RELATED WORK

Privacy concerns on the Android device primarily include two areas. One is permission management in the system. An Android app installed on the user's device will generally request permissions to acquire information from the user's device, which usually concerns the end user's privacy. Felt et al. suggest that developers may ask for unnecessary permissions due to confusion or forgetfulness[12]. Khatoon et al. indicate that malware apps will use permission to gather valuable user data. And even Android prompt users on an ask-on-first-use basis. Usually, an end-user is unclear about the accessibility of the permission granted to their personal information[20][7]. And a research in 2013 resolved that issue that most OS has no specific support to AR apps, thus granted excessive permissions and expose significantly additional information[19]. Some researchers have studied the feasibility of dynamically granting permission[39] or using machine learning to identify malware app permissions [32] as the solution to privacy protection.

The other one is the sensitive data leaks in Android apps. Many studies about data leaks exist, and tools have been developed to proceed with further research. Including the SUSI and Flowdroid employed in our study. Zhang et al. performed a case study and proposed a potential mechanism to detect unnecessary privacy leaks in MAR apps[40], which suggested the privacy issue in AR apps.

Our work is inspired by Roesner et al., who researched user security and privacy on wearable AR devices. They listed an initial set of challenges and defined a research agenda for security and privacy for AR systems[31][30]. There is a significant amount of research about the security and privacy of AR technology on wearable AR devices. For example, a study shows that AR headset displays can leak visual information to people or sensors near the user[21] . However, the amount of research about MAR apps is limited. Harborth et al. conducted a vignette-based experiment with 1,100 participants to analyze privacy concerns related to MAR apps, which indicated that the sensitivity of app permissions is one major factor that determines privacy concerns[17]. Nevertheless, they did not explore the actual permission setup of these apps. We believe it is necessary to look into the program of MAR apps currently in the app market.

Different from previous research, our study focused on the dangerous permissions that are at high risk of exposing user privacy, and we also investigated data leaks and data safety in MAR apps.

7 CONCLUSION AND FUTURE DIRECTIONS

In this paper, we present a preliminary empirical study on 390 real-world Android MAR apps to understand their private data practices, including the personal information they collect, whether the collected data is properly stored and processed, and the availability of their Data Safety.

We plan to continue our research in the following areas: **1) Sensor data analysis**. In order to support AR features, MAR apps heavily rely on sensor data such as *accelerometer*, *gyroscope*, *magnetic field*, *step counter*. We plan to investigate how sensor data is collected, stored, and processed by the MAR apps. Based on the sensor data leaks discovered in this study, we intend to investigate the techniques for detecting and preventing data leaks. **2) Permission analysis**. With the study results we have, we plan to perform further study to understand the context and purpose of those permission requests. **3) Data sharing of MAR apps**. Third-party libraries have been widely used in Android apps, such as libraries for crash analysis, advertising, and social platforms. The data sharing practices between third-party services and normal apps have been well studied, but not for MAR apps. We intend to find out which third-party services are used by MAR apps and how they share data with each other.

REFERENCES

[1] 2022. *AppBrain.* Retrieved July, 2022 from https://www.appbrain.com

[2] Kevin Allix, Tegawendé F. Bissyandé, Jacques Klein, and Yves Le Traon. 2016. AndroZoo: Collecting Millions of Android Apps for the Research Community. In *Proceedings of the 13th International Conference on Mining Software Repositories* (Austin, Texas) *(MSR '16)*. ACM, New York, NY, USA, 468–471. https://doi.org/10.1145/2901739.2903508

[3] Steven Arzt, Siegfried Rasthofer, and Eric Bodden. 2013. SuSi: A Tool for the Fully Automated Classification and Categorization of Android Sources and Sinks.

[4] Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau, and Patrick McDaniel. 2014. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. *Acm Sigplan Notices* 49, 6 (2014), 259–269.

[5] Adam J Aviv, Benjamin Sapp, Matt Blaze, and Jonathan M Smith. 2012. Practicality of accelerometer side channels on smartphones. In *Proceedings of the 28th annual computer security applications conference*. 41–50.

[6] Zhongjie Ba, Tianhang Zheng, Xinyu Zhang, Zhan Qin, Baochun Li, Xue Liu, and Kui Ren. 2020. Learning-based Practical Smartphone Eavesdropping with Built-in Accelerometer.. In *NDSS*.

[7] Kevin Benton, L. Jean Camp, and Vaibhav Garg. 2013. Studying the effectiveness of android application permissions requests. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. 291–296. https://doi.org/10.1109/PerComW.2013.6529497

[8] Pietro Cipresso, Irene Alice Chicchi Giglioli, Mariano Alcañiz Raya, and Giuseppe Riva. 2018. The past, present, and future of virtual and augmented reality research: a network and cluster analysis of the literature. *Frontiers in psychology* (2018), 2086.

[9] Tamara Denning, Zakariya Dehlawi, and Tadayoshi Kohno. 2014. In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2377–2386.

[10] Anthony Desnos. 2022. Androguard. Retrieved Augues 1, 2022 from https://androguard.readthedocs.io/en/latest/

[11] en.wikipedia.org. 2022. Augmented Reality. Retrieved July 31, 2022 from https://en.wikipedia.org/wiki/Augmented_reality

[12] Adrienne Porter Felt, Kate Greenwood, and David Wagner. 2011. The effectiveness of application permissions. In *2nd USENIX Conference on Web Application Development (WebApps 11)*.

[13] Google.Com. 2022. Manifest.permission. Retrieved July 31, 2022 from https://developer.android.com/reference/android/Manifest.permission

[14] Google.Com. 2022. Permissions on Android. Retrieved July 31, 2022 from https://developer.android.com/guide/topics/permissions/overview

[15] Google.Com. 2022. Provide information for Google Play's Data safety section. Retrieved Augues 1, 2022 from https://support.google.com/googleplay/android-developer/answer/10787469?hl=en

[16] David Harborth and Alisa Frik. 2021. Evaluating and Redefining Smartphone Permissions with Contextualized Justifications for Mobile Augmented Reality Apps. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. 513–534.

[17] David Harborth and Sebastian Pape. 2021. Investigating privacy concerns related to mobile augmented reality Apps – A vignette based online experiment. *Computers in Human Behavior* 122 (2021), 106833. https://doi.org/10.1016/j.chb.2021.106833

[18] Rakibul Hasan, David Crandall, Mario Fritz, and Apu Kapadia. 2020. Automatically detecting bystanders in photos to reduce privacy risks. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 318–335.

[19] Suman Jana, David Molnar, Alexander Moshchuk, Alan Dunn, Benjamin Livshits, Helen J Wang, and Eyal Ofek. 2013. Enabling {Fine-Grained} Permissions for Augmented Reality Applications with Recognizers. In *22nd USENIX Security Symposium (USENIX Security 13)*. 415–430.

[20] Asma Khatoon and Peter Corcoran. 2017. Android permission system and user privacy—a review of concept and approaches. In *2017 IEEE 7th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*. IEEE, 153–158.

[21] Tadayoshi Kohno, Joel Kollin, David Molnar, and Franziska and Roesner. 2015. *Display Leakage and Transparent Wearable Displays: Investigation of Risk, Root Causes, and Defenses.* Technical Report MSR-TR-2015-18. https://www.microsoft.com/en-us/research/publication/display-leakage-and-transparent-wearable-displays-investigation-of-risk-root-causes-and-defenses/

[22] Kiron Lebeck, Kimberly Ruth, Tadayoshi Kohno, and Franziska Roesner. 2018. Towards security and privacy for multi-user augmented reality: Foundations with end users. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 392–408.

[23] Sarah M Lehman, Abrar S Alrumayh, Kunal Kolhe, Haibin Ling, and Chiu C Tan. 2022. Hidden in Plain Sight: Exploring Privacy Risks of Mobile Augmented Reality Applications. *ACM Transactions on Privacy and Security* (2022).

[24] Xing Liu, Jiqiang Liu, and Wei Wang. 2015. Exploring sensor usage behaviors of Android applications based on data flow analysis. In *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 1–8.

[25] Xing Liu, Jiqiang Liu, Wei Wang, Yongzhong He, and Xiangliang Zhang. 2018. Discovering and understanding android sensor usage behaviors with data flow analysis. *World Wide Web* 21, 1 (2018), 105–126.

[26] Richard McPherson, Suman Jana, and Vitaly Shmatikov. 2015. No escape from reality: Security and privacy of augmented reality browsers. In *Proceedings of the 24th International Conference on World Wide Web*. 743–753.

[27] Emiliano Miluzzo, Alexander Varshavsky, Suhrid Balakrishnan, and Romit Roy Choudhury. 2012. Tapprints: your finger taps have fingerprints. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*. 323–336.

[28] Jack Nicas, Cat Zakrzewski, and J Greene. 2016. Augmented Reality Gets Boost From Success of Pokémon Go'. *Wall Street Journal* 13 (2016).

[29] Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Zhang. 2012. Accessory: password inference using accelerometers on smartphones. In *proceedings of the twelfth workshop on mobile computing systems & applications*. 1–6.

[30] Franziska Roesner and Tadayoshi Kohno. 2021. Security and Privacy for Augmented Reality: Our 10-Year Retrospective. In *VR4Sec: 1st International Workshop on Security for XR and XR for Security*.

[31] Franziska Roesner, Tadayoshi Kohno, and David Molnar. 2014. Security and privacy for augmented reality systems. *Commun. ACM* 57, 4 (2014), 88–96.

[32] Kavita Sharma and Brij B Gupta. 2019. Towards privacy risk analysis in android applications using machine learning approaches. *International Journal of E-Services and Mobile Applications (IJESMA)* 11, 2 (2019), 1–21.

[33] Julian Steil, Marion Koelle, Wilko Heuten, Susanne Boll, and Andreas Bulling. 2019. Privaceye: privacy-preserving head-mounted eye tracking using egocentric scene image and eye movement features. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. 1–10.

[34] Google Play Store. 2022. AR Apps. Retrieved Augues 1, 2022 from http://bit.ly/GooglePlay_ArApps

[35] Google Play Store. 2022. AR Games. Retrieved Augues 1, 2022 from http://bit.ly/GooglePlay_ArGames

[36] Xiaoyu Sun, Xiao Chen, Kui Liu, Sheng Wen, Li Li, and John Grundy. 2021. Characterizing Sensor Leaks in Android Apps. In *2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 498–509.

[37] Common weakness enumeration. 2006. CWE-532: Insertion of Sensitive Information into Log File. Retrieved Sept 1, 2022 from https://cwe.mitre.org/data/definitions/532.html

[38] Tobias Weyand, Ilya Kostrikov, and James Philbin. 2016. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*. Springer, 37–55.

[39] Primal Wijesekera, Arjun Baokar, Lynn Tsai, Joel Reardon, Serge Egelman, David Wagner, and Konstantin Beznosov. 2017. The Feasibility of Dynamically Granted Permissions: Aligning Mobile Privacy with User Preferences. In *2017 IEEE Symposium on Security and Privacy (SP)*. 1077–1093. https://doi.org/10.1109/SP.2017.51

[40] Xueling Zhang, Rocky Slavin, Xiaoyin Wang, and Jianwei Niu. 2019. Privacy assurance for android augmented reality apps. In *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*. IEEE, 114–1141.

[41] Leah Zhao, Neil Wong Hon Chan, Shanchieh Jay Yang, and Roy W. Melton. 2015. Privacy Sensitive Resource Access Monitoring for Android Systems. In *2015 24th International Conference on Computer Communication and Networks (ICCCN)*. 1–6. https://doi.org/10.1109/ICCCN.2015.7288451