JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 26 No. 6, pp. 107-113, June 2021
https://doi.org/10.9708/jksci.2021.26.06.107

Forgotten Permission Usages: An Empirical Study on App Description Based Android App Analysis

Zhiqiang Wu*, Scott Uk-Jin Lee*

*Student, Dept. of Computer Science & Engineering, Hanyang University, Ansan, Korea
*Associate Professor, Dept. of Computer Science & Engineering, Hanyang University, Ansan, Korea

[Abstract]

In this paper, we conducted an empirical study to investigate whether Android app descriptions provide enough permission usages for measuring app quality in terms of human writing and consistency between code and descriptions. Android app descriptions are analyzed for various purposes such as quality measurement, functionality recommendation, and malware detection. However, many app descriptions do not disclose permission usages, whether accidentally or on purpose. Most importantly, the previous studies could not precisely analyze app descriptions if permission usages cannot be completely introduced in app descriptions. To assess the consistency between permissions and app descriptions, we implemented a state-of-the-art method to predict Android permissions for 29,270 app descriptions. As a result, 25% of app descriptions may not contain any permission semantic, and 57% of app descriptions cannot accurately reflect permission usages.

▸ **Key words**: Android, App Description, Permission Semantics, Empirical Study, Natural Language Processing

[요 약]

본 논문에서는 안드로이드 앱 설명이 애플리케이션 품질 측정에 충분한 권한 사용을 제공하는지에 대해 연구하였다. 안드로이드 애플리케이션 설명은 품질측정, 기능추천(functionality recommendation), 말웨어감지와 같은 다양한 목적으로 분석된다. 그러나 많은 앱들이 설명에서 실수 혹은 고의로 권한 사용을 공개하지 않는다. 이전 연구에서는 가장 중요한 것은 애플리케이션 설명에서 권한 사용에 대한 내용이 없거나 부족하면 애플리케이션 설명을 정확하게 분석할 수 없었다. 권한과 앱 설명 간의 일관성을 평가하기 위해 29,270개의 애플리케이션 설명에 대한 안드로이드 권한을 예측하는 방법을 구현했다. 결과로 앱 설명의 25%는 권한에 대한 의미를 포함하지 않았으며 앱 설명의 57%는 권한 사용에 대한 내용을 정확하게 반영 할 수 없다.

▸ **주제어**: 안드로이드, 앱 설명, 권한 의미, 실증적 연구, 자연어 처리

• First Author: Zhiqiang Wu, Corresponding Author: Scott Uk-Jin Lee
*Zhiqiang Wu (wzq0515@hanyang.ac.kr), Dept. of Computer Science & Engineering, Hanyang University
*Scott Uk-Jin Lee (scottlee@hanyang.ac.kr), Dept. of Computer Science & Engineering, Hanyang University
• Received: 2021. 05. 11, Revised: 2021. 06. 08, Accepted: 2021. 06. 09.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

In recent years, mobile applications (apps) have become a part of people's daily lives, providing diverse functionalities to end-users such as mobile payment, communication, reaction [1]. Currently, Android is the most popular platform for mobile devices due to its open-source nature. In the first quarter of 2021, Play Store, the official app market, provides over 2.9 million apps for download. With numerous apps in the official market, developers provide vivid descriptions for apps [2]. Primary purposes for introducing an app with a striking description are (1) adhere to the policy of Google to disclose the usages of sensitive information that depend on the dangerous permissions [3], (2) attract user base with novel functionalities.

Therefore, several researches have been conducted to analyze the app descriptions in mobile domains, such as functionality recommendation, malware detection [23], quality maintenance. MPDroid [4] is applied for maintaining app quality, extracts the topic semantics from app descriptions and categorizes them into different clusters based on the latent topics. Then, MPDroid leverages collaborative filtering to filter a small set of permission for each cluster. Finally, sensitive API usage is grouped for each cluster to minimize the set of declared permissions for each cluster, which effectively assists developers in refining the permission usages and avoid the potential security risks. The main strength of MPDroid is that processes meta-information (i.e., app description and API usage in the manifest) to maintain permission usages. However, developers may preferentially introduce novel functionalities instead of common usages of dangerous permissions in order to promote apps due to limited characters in app descriptions [5]. If developers did not completely disclose usages of dangerous permissions, MPDroid is no longer valid.

For example, social network apps generally request *Camera, Storage*, and *Location* permissions to provide the corresponding functionalities, commonly occurring in this category. In the case of these common permissions based on category are unmentioned in the app description for unknown reasons. In that case, MPDroid may classify this app into another cluster and recommend incorrect permissions. Such a zero-tolerance policy for unmentioned permission in the app description facilitates developers to find out security risks. However, the existing techniques have never been considered unmentioned permissions, which cause large false positives on both functionality recommendation and maintenance [6].

To investigate this issue, we replicated the state-of-the-art technique FCDP [7] to predict the permissions from app descriptions. Then, we conducted an empirical study to discuss the coverage of permission semantics in app descriptions on large Android apps and descriptions from the Play Store. Based on the result, we provide novel insight to the Android research community.

The rest of the paper is organized as follows: Section II presents the related work of this study and proposes the research question. Then, Section III describes our research methodology. The evaluation results are presented in Section IV. Finally, we summarized our findings for app descriptions in Section V.

II. Preliminaries

1. Related works

In order to assist end-users in understanding whether dangerous permissions are imperative for their demands before installation, Play Store has issued a policy for writing app descriptions [7]. The policy indicates that developers should explicitly disclose the permission usages in the app descriptions if app requires accessing, sharing, collecting or using sensitive data from Android devices [17][22]. With this kind of market policy, app

descriptions contain lots of information such as permission usages, functionalities. Hence, app description analysis is extensively applied to detect consistency between app descriptions and permissions [8][21], maintain app quality [7][20], and extract novel functionalities from other apps [4].

WHYPER [9] is the first work to check whether the used dangerous permissions have been introduced in the app descriptions. Initially, WHYPER extracts the permission semantics from Android API documentation and identifies a set of key-words for each permission. Finally, if any sentence from the app description can disclose the declared permission, it is marked as benign. Otherwise, the usages of permissions are determined as suspicious. However, Qu et al. indicated that some APIs are undocumented with related permissions, which causes that WHYPER extracts incomplete semantics from API documentation [10]. Xiao et al. [4] proposed MPDroid to identify the minimum permission set for each topic based on app descriptions by using a collaborative filtering technique. In addition, SAFE was proposed to explore new features from other app descriptions to inspire developers as reference. Subaihin et al. [18] and Wu et al. [19] extract functionalities from app descriptions to analyze the similarity of apps and detect improper features respectively.

However, these studies explicitly present that app descriptions that contain at least one permission usages or features can be considered in the dataset. In other words, the existing approaches may not be applied for all apps in the market if the descriptions lack required semantics. Different from these researches, our study investigates whether the existing techniques are applicable for all apps and analyze the possible reasons why app descriptions cannot disclose all permission usages and meet the requirement of Google policy.

2. Research question

In this paper, we focus our analysis on the integrity of permission semantics in Android app descriptions. Any app in the Play Store can be the candidate in this study, whether its description disclose any permission. Specifically, we aim to address one research question that is concerned with unmentioned permissions in the app description:

RQ: Does any app description not mention all permission usages?

We reproduced the state-of-the-art technique to predict permissions from app descriptions to perform an empirical study on large Android apps from the Play Store.

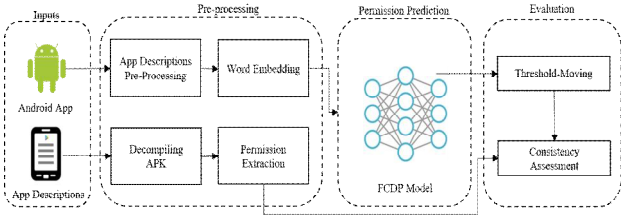

Fig. 1. Procedure of Methodology

III. Methodology

To answer this research question, we analyze the app descriptions from the Play Store and identifying relevant usages of permission in the code. To investigate the consistency between permissions and description, we conducted our experiments based on the existing permission prediction model as shown in Fig. 1. We extracted the permissions from app descriptions and the source code in Android apps, separately. According to basic pre-processing, the textual descriptions are converted into numerical vectors for training FCDP model. The declared permissions in source code are extracted by AndroGuard tool. Finally, we compare whether the declared permissions in source code are consistent with predicted permissions from app descriptions.

1. Data Collection

Following previous studies [7][11], we focus on Android apps with English descriptions. We randomly crawled over 100,000 app descriptions

from Play Store by using a Python script. To remove non-English descriptions from our dataset, we leveraged Compact Language Detector (cld3) [12]. After language filtration, we obtained 61,270 unique descriptions. Based on this, we collected Android apps from AndroZoo [13] that is the largest Android repository with more than 10 market sources. However, a given app has multiple versions in AndroZoo since it has been updated weekly. In order to ensure the consistency between the Android app and its description, the apps are extracted if they are updated within one month. Finally, the refined dataset consists of 29,270 apps.

Table 1. Predicted Permissions

Permission Groups	Permission
Calendar	WRITE_CALENDAR
Contacts	READ_CONTACTS
	WRITE_CONTACTS
	GET_ACCOUNTS
Location	ACCESS_FINE_LOCATION
	ACCESS_COARSE_LOCATION
Tasks	GET_TASKS
	KILL_BACKGROUND_PROCESS
Call Log	READ_CALL_LOG
Setting	WRITE_SETTINGS
Microphone	RECORD_AUDIO
Camera	CAMERA
Phone	CALL_PHONE
SMS	READ_SMS
	RECEIVE_SMS
Storage	WRITE_EXTERNAL_STORAGE

2. Permission Identification

Permission from descriptions. To investigate the permission semantics in the app descriptions, we reproduced FCDP [7] that achieved the better performance in predicting 16 dangerous permissions in 11 groups from app descriptions that is shown in Table 1. FCDP is a deepl earning model implemented with bidirection-Long Short-Term Memory (Bi-LSTM) with attention mechanism [16].

To train FCDP model, we selected a dataset that is provided by AC-Net as training dataset [15]. Following typical natural language processing, the training dataset has been removed stop words and stemming. Then, each sentence is assigned an 11-dimension vector which denotes 11 permission group we detected. If the sentence contains the corresponding permissions, the value in the vector should be 1. Otherwise, the value is 0, which denotes that the sentence does not contain this permission semantic.

In order to feed the textual sentences into deep learning model, Word2Vec is applied to embed the text into numerical vectors. After that, the numerical vectors are fed into FCDP model for training. As a result, one sentence may contain multiple permission semantics. For instance, the sentence "You may take a picture and save in your SD card" discloses two permissions (i.e., Camera and Storage). Finally, the permission semantics for an app can be aggregated by all predicted results of sentences. We consider that permission is depicted in the app description when at least one sentence is predicted with the corresponding permission. According to FCDP, each sentence is assigned a 11-dimension vector to represent whether the predicted sentence contain any permissions. We aggregate all sentences of the description to integrate a list of permission usage for the given app.

Permission from code. In the Play Store, not all apps request dangerous permissions for their functionalities. In this case, the permission semantics are not required in the app descriptions. Hence, we conducted a static analysis to confirm whether the app declared the permissions in the code. AndroGuard [14] is applied to extract the invoked permission in AndroidManifest.xml files. The experiments were conducted on a machine with an Intel Xeon E5-2698 CPU, 256 GB of RAM, and four NVIDIA Tesla V100 GPUs.

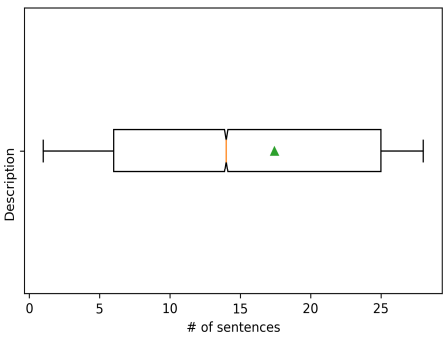

Fig. 2. Number of sentences in app descriptions

IV. Evaluation Results

This section presents our experimental results to reveal the effectiveness of app description in real-world apps. To answer the question, we focus on the following points in terms of the quality of human writing, the gap between app descriptions and invoked ones in the code.

1. Quality of Human Writing

The previous study indicates that the shorter descriptions have less chance to possess any permission semantic when the number of sentences less than 5 [7]. Therefore, we statistically analyze the number of sentences for all descriptions, as shown in Fig. 2. On average, each app description consists of 17.5 sentences, which may contain at least one permission. However, approximately 25% of app descriptions have less than 5 sentences, which may not contain any permission semantic.

To further justify whether the app description contains semantics, we performed FCDP to predict 61,270 app descriptions mentioned in the last section. We split these app descriptions into over 1.08 million sentences. After prediction, each sentence has been assigned labels to indicates its semantics. As shown in Fig. 3, only 17.62% of 1.08 million sentences explicitly/implicitly disclose the permission usages. Thus, app descriptions may not reflect all permission usages in the app.

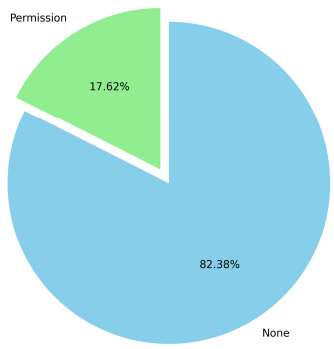

Fig. 3. Distribution of Permission Semantics

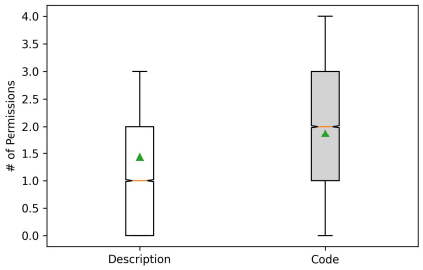

Fig. 4. Permission usages in app descriptions and code

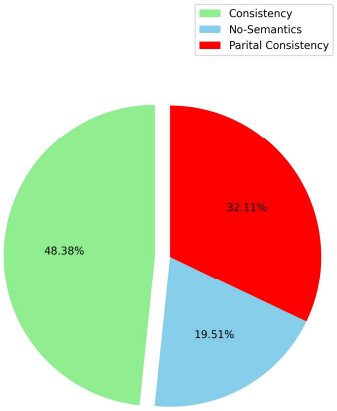

Fig. 5. Consistency between Permission and Code

2. Consistency between Descriptions and Code

In the app market, some apps just provide simple functionalities to users without using any permission. Therefore, we performed static analysis to investigate the difference between app descriptions and source code on 29,270 Android apps.

As shown in Fig. 4, our analysis shows that each app description discloses 1.43 permissions in general. However, apps invoked 1.86 permissions on average, which indicates that the app descriptions only disclose 76% of invoked permissions.

Fig. 5 shows the distribution of whether invoked permissions are completely depicted in the descriptions. As a result, 48.3% of app descriptions accurately reflect the permission usages. However, other 51.7% of app descriptions lack introducing at least one permission in the description. More specifically, 19.5% of app descriptions could not reflect any invoked permission.

V. Discussion

In this paper, we conducted an empirical study to discuss the descriptions in Android apps. Our analysis results indicate that the current app descriptions could not accurately reflect all permission usages in the code. Due to limited characters, developers only depicted the novel functionalities in the app descriptions in order to attract users, which also violates the policy of Google. On average, the app descriptions only disclose 76% of permission usages. Overall, the existing research is not applicable to all app descriptions in the market if the descriptions do not contain any permission usages.

App description as meta-information in the market has significantly contributed to measure the quality of apps and infer some novel features from app descriptions. The existing techniques did not consider such incomplete descriptions in real-world apps. In future research, we plan to combine other meta-information to extract the permission usages and features such as privacy policy, app category, and user reviews.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2020R1F1A1076208).

REFERENCES

[1] C. Zhang, H. Wang, R. Wang, Y. Guo and G. Xu, "Re-checking App Behavior against App Description in the Context of Third-party Libraries," Proceeding of International Conference on Software Engineering and Knowledge Engineering , 2018. DOI: 10.18293/SEKE2018- 180

[2] A. A. Subaihin, F. Sarro, S. Black and L. Capra, "Empirical Comparison of Text-based Mobile Apps Similarity Measurement Techniques," Empirical Software Engineering, vol.24, pp. 3290-3315, 2019.

[3] Privacy, deception and device abuse. Available: https://support.google.com/googleplay/android-developer/topic/9877467

[4] J. Xiao, S. Chen, Q. He, Z. Feng, and X. Xue, "An android application risk evaluation framework based on minimum permission set identification," Journal of Systems and Software, vol. 163, pp. 110533, May 2020.

[5] L. Yu, X. Luo, C. Qian, and S. Wang, "Revisiting the description-to-behavior fidelity in android applications," Proceedings of IEEE International Conference on Software Analysis, Evolution, and Reengineering, pp. 415-426, 2016.

[6] M. Shamsujjoha, J. Grundy, L. Li, H. Khalajzadeh, and Q. Lu, "Checking app behavior against app descriptions: What if there are no app descriptions?" Proceedings of International Conference on Program Comprehension, pp. 422-432, 2021.

[7] Z. Wu, X. Chen, and S. U. J. Lee, "FCDP: Fidelity Calculation for Description-to-Permissions in Android Apps," IEEE Access, vol. 9, pp. 1062-1075, Jan. 2021.

[8] T. Watanabe, M. Akiyama, T. Sakai, and T. Mori, "Understanding the inconsistencies between text descriptions and the use of privacy-sensitive resources of mobile apps," in Proceedings of Symposium On Usable Privacy and Security, pp. 241-255, 2015.

[9] R. Pandita, X. Xiao, W. Yang, W. Enck and T. Xie, "WHYPER: Towards automating risk assessment of mobile applications," Proceedings of USENIX Security Symposium, pp. 527-542, 2013.

[10] Z. Qu, V. Rastogi, X. Zhang, Y. Chen, T. Zhu, and Z. Chen, "Autocog: Measuring the description-to-permission fidelity in android applications," Proceedings of ACM Conference on Computer and Communications Security, pp. 1354-1365, 2014.

[11] A. Gorla, I. Tavecchia, F. Gross, and A. Zeller, "Checking app behavior against app descriptions," Proceedings of International Conference on Software Engineering, pp. 1025-1035, 2014.

[12] Compact language detector, https://github.com/google/cld3

[13] K. Allix, T. F. Bissyand´e, J. Klein, and Y. Le Traon, "Androzoo: Collecting millions of android apps for the research community," Proceedings of International Conference on Mining Software Repositories, pp. 468-471, 2016.

[14] AndroGuard, https://github.com/androguard/androguard

[15] Y. Feng, L. Chen, A. Zheng, C. Gao and Z. Zheng, "AC-Net: Assessing the Consistency of Description and Permission in Android Apps," IEEE Access, vol. 7, pp. 57829-57842, Apr. 2019.

[16] H. Alecakir, M. Kabukcu, B. Can and S. Sen, "Attention: there is an inconsistency between android permissions and application metadata!," International Journal of Information Security, pp. 1-19, Jan. 2021.

[17] Y. Hu, H. Wang, T. Ji, X. Xiao, X. Luo, P. Gao and Y. Gao, "CHAMP: Characterizing Undesired App Behaviors from User Comments based on Market Policies," Proceedings of IEEE/ACM

International Conference on Software Engineering, pp. 933-945, May 2021.

[18] A. A. Subaihin, F. Sarro, S. Black and L. Capra, "Empirical Comparison of Text-based Mobile Apps Similarity Measurement Techniques," Empirical Software Engineering, vol. 24, pp. 3290-3315, Jun. 2019.

[19] H. Wu, W. Deng, X. Niu and C. Nie, "Identifying Key Features from App User Reviews," Proceedings of International Conference on Software Engineering, pp. 922-932, May 2021.

[20] F. H. Shezan, K. Cheng, Z. Zhang, Y. Cao and Y. Tian, "TKPERM: Cross-platform Permission Knowledge Transfer to Detect Overprivileged Third-party Applications," Proceedings of Network and Distributed Systems Security Symposium, pp. 1-15, 2020.

[21] O. Olukoya, L. Mackenzie, I. Omoronyia, "Security-oriented View of App Behavior using Textual Descriptions and User-granted Permission Requests," Computers & Security, vol. 89, pp. 1-18, Feb. 2020.

[22] Z. Wu, X. Chen and S. U. J. Lee, "Permissions based Automatic Android Malware Repair using Long Short-Term Memory," Proceedings of the Korean Society of Computer Information Conference, pp. 387-388, 2019.

[23] Z. Wu, X. Chen and S. U. J. Lee, "Identifying Latent Android Malware from Application's Description using LSTM," Proceedings of International Conference on Information, System and Convergence Applications, pp. 40-42, 2019.

Authors

Zhiqiang Wu received the B.S. in computer science from Shanghai Polytechnic Univ. in 2015. He also received M.S. degree from Hanyang University in 2017. Currently, he is pursuing the Ph.D. degree in computer science with the Dept. of Computer Science & Engineering, Hanyang University, Ansan, South Korea. His research interests include Android apps analysis, code smells on security and software refactoring on mobile apps.

Scott Uk-Jin Lee received the B.S. degree in software engineering and the Ph.D. degree in computer science from University of Auckland, New Zealand. He was a Post-Doctoral Research Fellow at the Commissariat à l'énergieatomique et aux énergies alternatives, France. He is currently serving as Associate Professor of the Department of Computer Science and Engineering, Major in Bio Artificial Intelligence. His research interests include software engineering, formal methods, and quality assurance. He is also a member of the Korean Institute of Information Scientists and Engineers and the Korean Society of Computer and Information. He has served as an editor, the technical chair, and a committee member for several journals and conferences.