Global Scientific JOURNALS

GSJ: Volume 9, Issue 9, September 2021, Online: ISSN 2320-9186
www.globalscientificjournal.com

PERMISSION ANALYSIS OF MOBILE APPLICATIONS

Abdullah Khalil

Department of Computer Science & Information Technology, Univerity of Engineering & Techology Peshawar, Pakistan
Email: abdullah.khalil@yahoo.com

KeyWords
Permissions; Security Analysis; Mobile Applications

ABSTRACT

Security Analysis is the process of evaluating an application's vulnerabilities against a given set of exploits. It gives an insight of the application's working, requisite permissions, third-party dependencies, external API calls etc. In our perspective, security analysis is used to find vulnerable points and possible abnormal behavior based on a given random / fuzzy input which may aid in preparing possible attack vector for cracking the application. Purpose of the study is to have an insight of the applications' behavior in terms of code de-obfuscation, grant of access for changing the application's code and assessing the permissions as ample or beyond the needs of the mobile application under study.

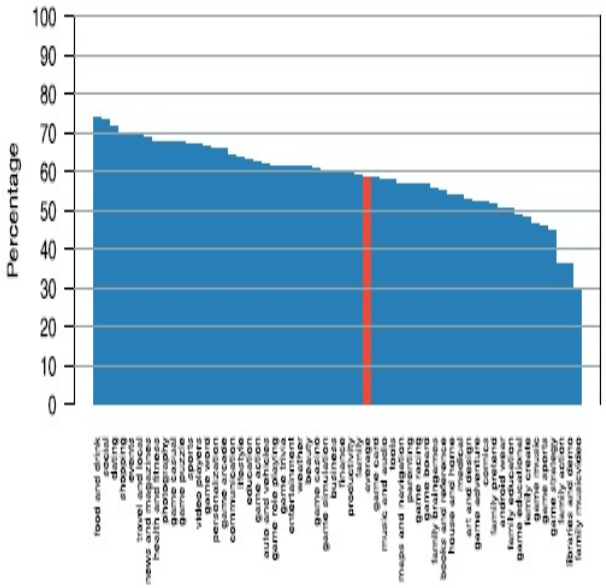
INTRODUCTION

The aim of this research is to determine the degree of privacy provided by and abused by mobile apps. This is accomplished by reverse engineering these programs. Since mobile is still a nascent medium, reverse engineering mobile applications is limited. Many methods do not have in-depth insight or require the use of several tools to accomplish the goal. Furthermore, almost all industries use mobile apps to expand their offerings because it is an evolving, common, and oriented medium. Mobile apps are security-sensitive applications, and even the tiniest flaw may have a significant effect on both the consumer and the industry.

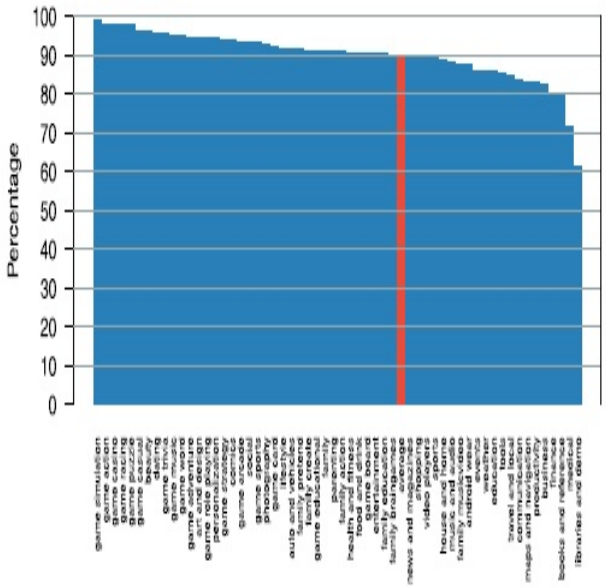
LITERATURE REVIEW

Mobile applications have been used in many aspects of life including financial as well as healthcare, therefore, it is important to make such applications secure. Current mobile operating systems (OS) depend heavily on the authorization-based security model to implement operational constraints that each application can execute [1].

A number of authors have done reverse engineering of mobile-based applications, for example [2], but they have failed to extract permission-based anomalies from the same applications. Studies as in [3] have been conducted proposing the implementation of anti-debugging (AD) and anti-tampering (AT) protections, however the scope of their application by app developers is unknown. A survey in [3], for example, examined 14,173 apps from 2015 and 23,610 apps from the Google Play Store from 2019, the findings were very astounding. However, half of the remaining apps only implement one defense and do not utilize the variety of protections available. In the report, however, it was promising to note that apps employ more reverse engineering security in 2019 as compared



(a) Percentage of Apps Implementing at Least One AD Protection



(b) Percentage of Apps Implementing at Least One AT Protection

to apps in 2015 [3].

Although several studies have been conducted for the purpose of reverse engineering and visual exploration of android applications [4], there is still a gap in the identification of permission-based access authorization. In Android, it is not very easy to run software analysis on the executable code [5].

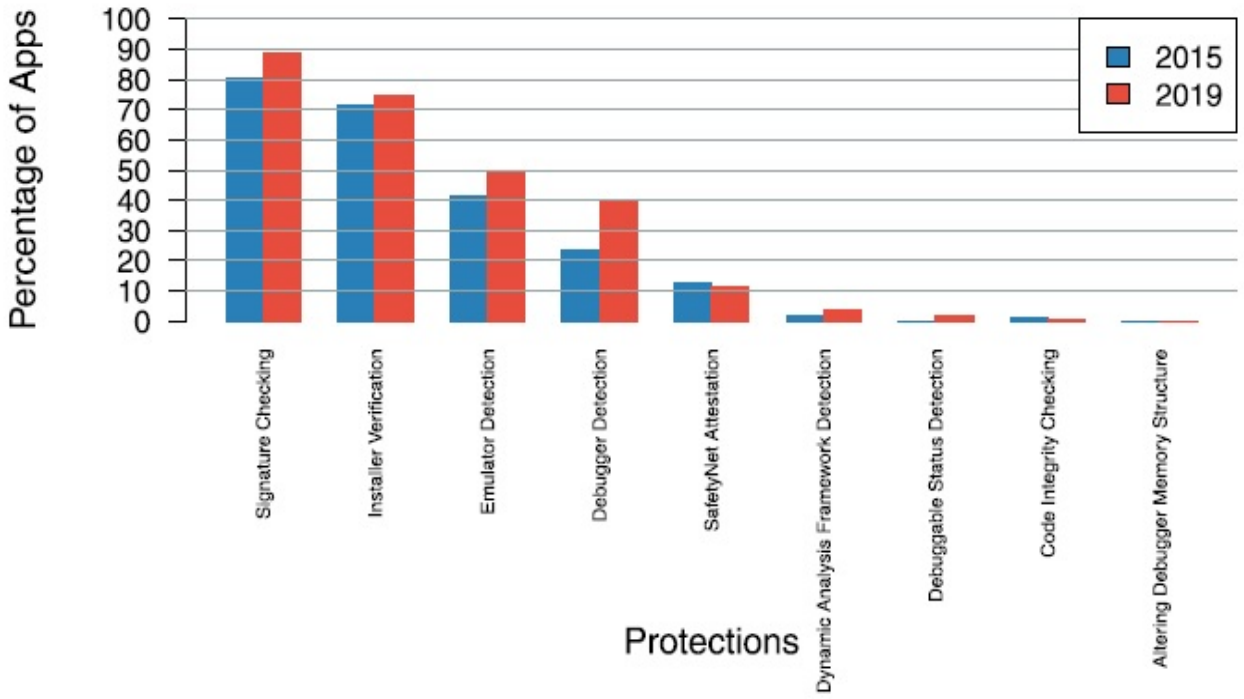


Figure 2.2: Apps statistics having security incorporations

RESEARCH METHODOLOGY

The approach that was used to accomplish the goal included the following milestones.

1. Literature Review
2. Data Collection
    a. Tools Collection
    b. Apps Collection
3. Apps Classification
    a. Worldwide
    b. Pakistan
    c. Shortlisting Apps for reverse engineering
4. Reverse Engineering for ascertaining permissions granted
    a. Code Analysis / Permission analysis using GUI Tools
5. Findings
6. Report documentation

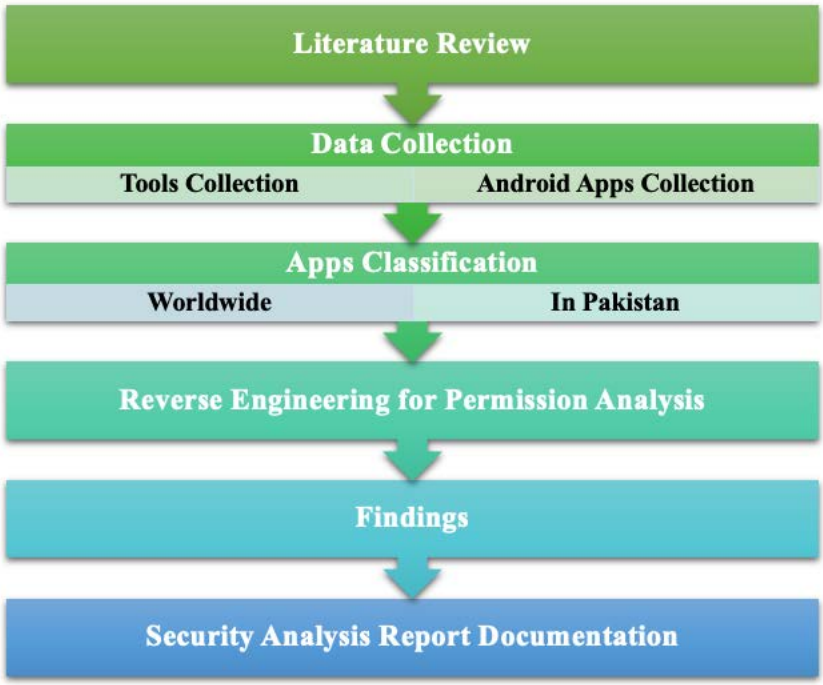


Figure 3.1: Process Flow Chart

Conclusion

In this paper we analyzed the most commonly used mobile applications for permission analysis and categorized them in various categories. The technique of static analysis is used and reverse engineering is also utilized. The application behaviour is studied through obfuscation. The applications are grouped into safe and unsafe categories. The results showed more than 50% of the anlayzed applications are unsafe interms of permission analysis.

References

[1] E. K. S. M. D. J. Hamid Bagheri, "A formal approach for detection of security," Formal Aspects of Computing, November 2017.

[2] W. Z. Zarni Aung, "Permission-Based Android Malware Detection," INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH, vol. 2, no. 3, March 2013.

[3] M. C. Stefano Berlato, "large-scale study on the adoption of anti-debugging and anti-tampering protections in android apps," Journal of Information Security and Applications, 2020.

[4] H. G. N. S. Ashutosh Jain, "Enriching reverse engineering through visual exploration".

[5] L. W. N. M. N. A. C. S. YAUHEN LEANIDAVICH ARNATOVICH, "A Comparison of Android Reverse Engineering Tools via Program Behaviours Validation Based on Intermediate Languages Transformation," IEEE Access, 2017.

[6] H. K. a. J. H. Y. Taejoo Cho, "Security Assessment of Code Obfuscation Based on Dynamic Monitoring in Android Things," IEEE ACCESS, 2017.

[7] F. M. I. M. M. P. Gianluca Dini, "Risk analysis of Android applications: A user-centric solution," Future Generation Computer Systems, Vols. S0167-739X(16)30153-4, no. https://linkinghub.elsevier.com/retrieve/pii/S0167739X16301534, 2016.

[8] W. H. Y. L. Zheran Fang, "Permission based Android security: Issues," ELSEVIER, pp. 1-14, 2014

"webarx," [Online]. Available: https://www.webarxsecurity.com/5-reasons-website-security-important-2018/. [Accesed 25 10 2019].