A Study of Security Threats in Android Applications from the Health and Fitness category

Aniqa Zaida Khanom
azkhanom@ncsu.edu

May 01, 2020

Abstract

With the ever-increasing use of smartphone technology, the use of mobile applications has gained many attractions in the field of healthcare and fitness in improving personal wellness. However, since these applications are also generating and storing sensitive data of the users every minute, the privacy and security challenges have also increased. This project explores the security threats in the top 500 Health and Fitness applications from Google Play Store based on the OWASP Top Ten Mobile Security Risks and tries to find a correlation between the security of an app with their popularity and usage. The project expects to find how the security risks change, or if they change at all when the ranking of the apps are considered. The initial hypothesis was that highly used mobile apps would have less severe vulnerabilities since they tend to update and fix bugs more frequently. However, this paper concludes that the hypothesis is partially correct even though no concrete statement could be made according to the results. Ultimately, this paper tries to infer in what ways the security threats differ in Android applications and how important the role of design and implementation of an application plays in its privacy and security risks.

1 Introduction

In a little over a decade, mobile applications have managed to become an integral part of our daily lives. This increasing capability of mobile technology offers many opportunities to improve health and wellness primarily by enabling healthcare that is more accessible, affordable and available [22, 2]. The same technology, however, could cause users harm if the hardware and software systems are not designed with security and privacy in mind. When security and privacy in mobile apps are considered, it is vital that both user-side and server-side infrastructure is brought into account. The OWASP Mobile Top 10 is a project which has been developed to categorize and provide resources to maintain security and privacy in mobile apps. It helps in detecting the threats and reduce their impact and is not only based on the end user device security, but also on the server-side infrastructure [27].

While there are many app stores, Google Play is the leading app store for mobile apps with about 3.3 millions of applications, with over 97 thousand apps under Health and Fitness. However, only about 5 percent of these apps have more than 50 thousand downloads, which clearly points to how most of these apps are not very usable. There have been instances where the apps were found to be shady and not very useful. An investigative report in 2012 found that out of 1500 health apps studied, 1 of 5 claimed either to treat or to cure a wide range of medical problems by just using the light, sound, or vibrations of the phone [40]. There have been several other new articles and reports which have claimed that Android apps that have been downloaded more than 100 million times were found to be infected with a malicious ad library that secretly distributes spyware to users and can perform dangerous operations. [21].In 2015, IBM reported that over 11.6 million mobile devices have been affected by malicious attacks [20] . The most frequently attacks in mobile apps are: Code Injection and Cross-Site Scripting (XSS).

Mobile health (mHealth) is one of the growing and most researched fields in the current world. With the advancement of mobile technology, it has become easier to access healthcare using various apps. However, since health and fitness apps usually contain a lot of sensitive data, starting from where the users and when they are engaging in fitness activities to their sleep schedule and period cycles. Prior work has shown that many third-party apps access information for which they do not have a legitimate need; some forward information to providers without the user's knowledge or consent [18]. There have been a number of research works done which test the security of sets of mobile health applications [44, 35, 17]. However, in most of these researches, the

number of apps that have been considered is pretty low, compared to the number of apps that are available; most of the papers consider 30-200 apps. Additionally, there are a lot of surveys and papers which focus more on the guidelines and legal aspects concerned with data protection regulations [37].

This paper explores the security threats in the top free 480 Health and Fitness applications from Google Play Store based on the OWASP Top Ten Mobile Security Risks and tries to find a correlation between the security of an app with their popularity and usage. The primary hypothesis was that the popular and highly used applications would have fewer vulnerabilities compared to the apps which are less popular and not implemented properly. In this paper, the APK files of 480 top free applications in Health and Fitness were collected using a python script, GooglePlayCrawler [23]. It supports batch downloading by collection/categories as well as recording the metadata of downloaded APKs. Using Static Analysis tools like Kiuwan and Quick Android Review Kit (QARK), the vulnerabilities of these apps were detected and analyzed.
This paper tries to make the following contributions:

- Explores the OWASP Mobile Top 10 security risks
- Correlation between security and the rank of an Android application
- A qualitative comparison between security tools Kiuwan and QARK based on their detection of true-positive vulnerabilities.

The remainder of this paper proceeds as follows: Section 2 contains the overview of the project, including data collection and the tools that were used. Section 3 evaluates and compares the results of the tool. Section 4 is a discussion of what the results infer. The subsequent section includes some of the related work done in the field of mobile Health (mHealth). The last section, Section 6, contains conclusion and future work on this project.

2 Overview

This project explores the security vulnerabilities in the top 480 free Android applications from Google Play Store. The method of work is pretty straight-forward and intuitive. To download batch APK files, python script GooglePlayCrawler was used, which can be found in GitHub. Using the script, the dataset, APK files of the mobile apps were collected from Play Store. Afterwards, two different security tools, Kiuwan and Quick Android Review Kit (QARK), was used to analyze and find vulnerabilities in the APK files. While one of the study goals was to compare the results received from the two static analysis tools, the target could not be fulfilled since Kiuwan is a paid platform and the trial version allows to analyze fifty applications only. Hence the comparison between the two security tools could not be done. Lastly, to compare the ranks of certain apps with their security vulnerability report, the metadata of the apps were used. The subsequent subsections describe details about the dataset collection and pre-processing and the security tools used in this project.
Figure 1 shows the approach taken in this paper at a high-level.

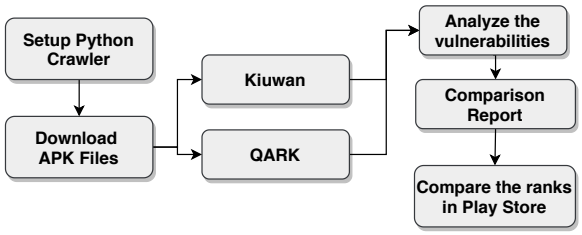

Figure 1: A high-level overview of the project

2.1 Data Collection

A large portion of the work in this section was to set up the environment for the script to work. GooglePlayCrawler [23] is a python script, developed by Duling Lai, that can batch download mobile apps from Google Play Store. It supports batch downloading by collection/categories (default to Top Free 120) along with recording the information (such as downloads/version/size/ratings/supported Android version etc.) of downloaded APKs in a csv files. It depends on a few unofficial Google Play APIs to work properly. All of these files, along with their usage and documentation, can be found in GitHub.

- Google Play Scraper: It is a Node.js module to scrape application data from the Play Store. This is used to collect the metadata of the APK files which are later used to compare ratings with the security threats found. [34]
- Google Play API: This module turns the Google Play Scraper into a RESTful API. [33]
- GPlayCli: It is a command line tool to search, install and update Android applications from the Google Play Store. It requires the user to have an authenticated login by either the use of tokens or credentials. For this project, the default feature of fetching a token from their assigned token dispenser server was used. [29]

Once all the files have been downloaded and installed, the Python script was run four times, downloading 120 apps each time, from the Top-Free list of Health and Fitness apps in the Play Store. It is worth mentioning that Google Play Store only shows up to 500 applications for every categories. Unless specifically searched for, the rest of the apps cannot be found. Hence, the crawler is limited to downloading 500 apps at most, given a specific category and collection. The crawler also had four different csv files, which contained the metadata of the apps. They were merged into one csv file for the ease of comparison in the later steps.

2.2 Security Tools

2.2.1 Kiuwan

Kiuwan is a static application security testing and source code analysis platform that offers state-of-the-art security and advanced business analytics [5]. It is a multi-technology platform that provides objective data reports for each application, and provides actionable guided plans to remediate the security defects and quality of all deliveries. Since it has the largest technology coverage for mobile applications currently, they can possibly detect more vulnerabilities than other free security testing tool.

The platform provided a two-week student trial for this project. However, only about fifty apps were scanned using this platform before the trial ended. As a result, while this might be a good tool, it was not appropriate for this research project.

2.2.2 QARK

Quick Android Review Kit (QARK), developed by LinkedIn, is a tool capable of finding common security vulnerabilities in Android applications, either in source code or packaged APKs [24]. Unlike commercial products, it is free to use and features educational information allowing security reviewers to locate precise, in-depth explanations of the vulnerabilities. QARK automates the use of multiple decompilers, leveraging their combined outputs to produce results, when decompiling APKs. Finally, it also points to possible vulnerabilities. It creates a report in JSON format for each application. In the author's opinion, this static analysis security tool was a good pick for this research since all of its functionalities can be used. The source code of the tool can be found in Github and the installation process is rather simple as well.

Android Studio: QARK can generate a basic exploit APK for a few of the vulnerabilities that have been found, by generating ADB commands.To generate the exploit APK, Android SDK is required. This feature is helpful in determining whether a detected vulnerability is false-positive or not by trying to exploit the vulnerability in Android Studio. Since this project deals with 480 applications, it was not possible to use this feature to test whether the detected threats were indeed exploitable vulnerabilities. However, that just points to potential future work for this project.

3 Qualitative Analysis

3.1 OWASP Mobile Top 10

Open Web Application Security Project, or OWASP, runs several community-run open-source projects to improve the knowledge of security enthusiasts all over the globe. One of their projects, launched in 2014, is the Top 10 Mobile Security Risks, as per OWASP Top 10 Project. These risks are the most common and exploitable risks when it comes to any mobile application. The risks, based on the latest list made in 2016 are [27]:

M1: Improper Platform Usage
M2: Insecure Data Storage
M3: Insecure Communication
M4: Insecure Authentication
M5: Insufficient Cryptography
M6: Insecure Authorization
M7: Client Code Quality
M8: Code Tampering
M9: Reverse Engineering
M10: Extraneous Functionality

The goal of this project was to categorize the vulnerabilities found through static analysis based on the above mentioned risks. These risks cumulatively covers user-side and server-side infrastructures, data security, network security, crypto-protocols and attacks and defense mechanisms.

3.2 Results

The static security tool used in this project generated a report in JSON format for each of the mobile applications. Owing to the limitation in time, all the reports could not be read and analyzed at the time of writing this paper. However, the first hundred apps and the last hundred apps, based on the ranking, were looked into to see what kinds of vulnerabilities these apps have, and if there is a trend that can be observed. While all the vulnerabilities reported for the apps observed did not follow any trend per se, however, most of the vulnerabilities were from related to M1(Improper Platform Usage), M2(Insecure Data Storage), M3(Insecure Communication), M5(Insufficient Cryptography) and M8(Code Tampering). Of these, the biggest security threat which was present in most

of the applications was Insecure Communication and Insufficient Cryptography. There were many instances of hardcoding sensitive information and passing sensitive information without proper encryption. It was expected to see many vulnerabilities relating to code tampering, but it was less present than what was expected to be. It can be inferred that code tampering is not as misused in mobile applications as in web applications. Further, almost all the applications had a good security against M4, Insecure Authentication, and M6, Insecure Authorization, and there were no instance found which could be mapped to Reverse Engineering. The last risk, Extraneous Functionality was a little complicated to map to the vulnerabilities since it deals with a wide spectrum of things. Comparing the first 100 apps with the last 100 apps, there wasn't any security risk from the above list which was present in only one of these two sets.

When the number of vulnerabilities were considered, it was found that the first set of apps have less number of vulnerabilities compared to the other set of apps. However, when the ranks are considered between the two sets, there isn't any fixed trend. For instance, the sixth app, Google Fit, had 4 less vulnerabilities than the third app, Period Tracker.However, Runtastic by Adidas, which is ranked seventh, had the same number of vulnerabilities as Period Tracker. This brings up a vital finding, which might be a little intuitive. Since the ranks are based on the ratings and the number of active users, it points out how most users are not aware of the security threats they're facing by using an app. Even an app without proper security measures may have a higher rank only because it has more users for some usability aspect. Additionally, it was observed that applications developed from bigger and more structured companies usually have less vulnerabilities. The Period Tracker app, while is used actively by many users, isn't updated as frequently as Google Fit is, even though the number of users is less for Google Fit. This may play a role in the number of vulnerabilities. This resonates with the initial hypothesis that the frequently updated apps might have lesser vulnerabilities.

All said, only the number of vulnerabilities were considered while comparing the apps for the ease of comparison. However, it is not a very effective metric. The severity of the vulnerabilities are also required to be brought to attention while differentiating the security risks of the apps since the severity defines whether a vulnerability is minor or needs immediate attention to get fixed. Further, a disadvantage of static analysis is the number of false positives in detected vulnerabilities. Usually, static analysis reports vulnerabilities which might not be exploitable by adversaries at all. QARK provides a nice functionality of creating ADB commands which can be used to exploit applications through Android Studio. It can verify whether a vulnerability is false positive or truly exploitable through this. However, due to time limitation, this is another prospective future work.

4 Discussion

The goal of this research project was to explore how the security threats in Android Applications from the Health and Fitness category. A few assumptions were made for this project. For this project, it was assumed that the app developer and the company are a part of the trusted computing base. On the other hand, the adversaries were considered to be third-party companies who collect data, somebody who wants to keep track of the user or wants to get payment information and exploit it. The goals of the adversaries are to collect, modify, fabricate or interrupt user data. One of the motivation of collecting these data might be to sell them to advertisers without the consent of users. The focus of this project was to understand how the vulnerabilities that are detected by the tools can be exploited by attackers. Ultimately, the goal was to determine if the ranks of the Android apps in Play Store have any relation to the number of security threats or not. It is intuitive to some extents that the applications which are being updated and having bug fixes more often have less vulnerabilities compared to the rest. However, at a higher level, it can be said that the ranks have nothing to do with the security vulnerabilities that exist in the applications. However, depending on the company the apps are developed by and their structure effects the quality of the apps. For instance, in general, apps developed by Google had less vulnerabilities compared to the apps by a small company. Additionally, even though 480 apps were scanned using QARK, due to limited time, all the reports could not be analyzed in detail.

If the novelty of this paper is brought into account, there has been similar work done before in the last few years. However, in most of these projects, irrespective of the categories of the apps, a small number of apps, usually between 30-100 apps were considered. In this project, since there were 480 applications, the primary challenge was to read through all the reports generated after the static analysis. As a result, it can be inferred that proper investigation of the reports could provide insights which haven't been included here. Secondly, one of the study goals was to compare an open source SAST tool, QARK, with a paid SAST tool, Kiuwan, to see which performs better. However, since the student trial of Kiuwan could not perform static analysis on more than fifty APK files, the comparison could not be done.

mHealth apps have become the third-most popular category of Android apps in the last few years. The invention and development of wearable devices like smart watches, ECG monitors, blood pressure monitors, etc have made things easier for the users, but have simultaneously made security and privacy a vital issue. While the apps have been improving when it comes to security and privacy in mobile apps, it is far behind than where it needs to be. There are active researches working to finding better security solutions everyday. Hence, the field of research in mHealth applications is only ever-increasing. A point worth mentioning is that since more work is being done concerning Android applications, and working with iOS applications is rarer, there is much more that can be done to ensure security and privacy in iOS applications.

When it comes to this paper, since most of the initial study goals and research questions remained unanswered in this paper, the possibilities of future work is endless. In the immediate future, the simplest task could be a proper analysis of all the reports that have been generated for this project. The functionality of QARK that can exploit the APK files using ADB commands in Android Studio to verify if the detected vulnerabilities are true-positive or not wasn't used in this project. Additionally, there are other static analysis tools and which correlate with The OWASP Mobile Top 10 more closely, like the Mobile Application Security Testing tool by Synopsys. Further, dynamic analysis can be done since it has less false-positive cases and the vulnerabilities detected by DAST tools are usually exploitable. The information flow in these apps were not looked at in this paper either. Hence this project could be the starting point to explore usability and privacy in this set of apps.

5 Related Work

At the beginning of this project, the study was concerned with the privacy issue in mobile Health apps. Since the scope of mobile applications has invaded all fields in a very short time, the security and privacy concerns have multiplied significantly. While researches related to mobile applications have started in the last decade, the field of research for mHealth apps started a few years late. However, as of now, mHealth apps and the security and privacy concerns have gained the peak of interest in researchers and a lot of different works have been done in the last few years. One of the most prominent and pioneering works in mHealth, was done by David Kotz, Carl A. Gunter et al. which sparked the first interest of research in this field [22]. It gave a fresh perspective of the privacy and security concerns related to mobile health applications and the writers have pointed out how the rapid increase of use of smartphones and tablets by patients, caregivers, and healthcare providers for health and wellness applications is one of the six major trends that pose security and privacy challenges in the current world. Research in this field further gained momentum since the development of wearable technology, like smart watch and medical devices, specially since they are collecting data of the users every second.

Between the last five years, a number of works have been done relating to the privacy and security concerns of mobile health applications. Privacy and security in mobile health apps: a review and recommendations is a paper from 2015 that summarizes the legal aspects of security and privacy in three parts: a study of the existing laws regulating these aspects in the European Union and the United States, a review of the academic literature related to this topic, and a proposal of some recommendations for designers in order to create mobile health applications that satisfy the current security and privacy legislation [28]. When privacy is concerned, a lot of laws and policies come into the picture. An article published by Privacy Rights Clearinghouse in 2013 analyzed the privacy in 43 Android and iOS health and fitness apps, and drew conclusions based on the laws of The United States of America [4]. Similar to the above paper, there have been a number of other works which focus on the legal aspects, current malpractices or the lack of practice thereof, privacy policies of the companies, compliance with the different security and privacy guidelines, etc [35, 38, 1, 36]. A project by Sunyaev(2015) assessed the availability, scope, and transparency of mHealth app privacy policies on iOS and Android, concluding that the available privacy policies are not transparent to users, require college-level literacy, and are often not focused on the app itself [43]. Another project co-authored by Sunyaev establishes an overview of mHealth apps offered on iOS and Android with a special focus on potential damage to users through information security and privacy infringements [9].There have been a significant number of works which were directed towards the data flow and leaks of the sensitive data through applications and how these data leaks might be prevented [18, 14]. In a paper by a group of researchers, they described how consumer data generated from mobile health apps might be distributed and reused [14]. Additionally,they outlined risks to individual privacy and security presented by this potential for aggregating and combining user data across apps.In another paper by the same authors, they did a systematic review to describe prevalent and emerging methods for searching, data extraction, and analysis in the context of mobile health-related apps targeted at consumers in a commercial app store [15].

With the advancement in health and fitness in mobile applications, there has been projects introducing new platforms and wearable techs which resonates with the idea of using mobile applications to make healthcare more affordable and available. In a paper by Gay and Leijdekkers, they introduce their research of 8 years by proposing a new health and fitness app, myFitnessCompanion, which enables users to aggregate their data in one place [13]. It discusses the technical challenges that have to be encountered to aggregate health and fitness data using a mobile device to enable interoperability. In [6], by collecting bioimpedance samples using a small wearable device we designed, our system can determine that (a) the wearer is indeed the expected person and (b) the device is physically on the wearer's body. It is claimed that their recognition method works with 98 percent balanced-accuracy under a cross-validation of a day's worth of bioimpedance samples from a cohort of 8 volunteer subjects. One paper talks about the current lack in security of smartphone-linked wearable sensors and point-of-need diagnostic devices built around real-time data streams which will enable care and enhance our understanding of physiological variability [42]. There are papers which explore the information leakage in wearable apps [8, 32, 7].

Diverting to mobile security, within the last decade, there have been a number of projects which focus this topic. Some of these projects resonates with the work intended to do in this paper. An article in 2015 identified a set of risk and safety features for evaluating mHealth apps and conducted a comparative analysis of the 10 most popular mHealth apps in Android and iOS platform each [39]. In [3], the authors analyzed the vulnerabilities detected in mobile medical apps according to risk factor standards defined by OWASP in 2014. [26] provides an overview of automatically testing the security and robustness of Android apps running on the cloud, a technique capable of generating and executing a large number of test cases for fuzzing an app. In a thesis dissertation by Dongjing He, a student from University of Illinois in 2014, a three-stage study of the mHealth apps to show that mHealth apps make widespread use of unsecured Internet communications and third party servers was presented [17]. The study deals with the prevalence of mobile app threats in mHealth Android apps. It further explores if there is any limitation in fundamental Android security design that can be used by malicious parties to disclose users' sensitive information. In [19], the paper presents a conceptual framework to improve the security of medical data associated with Android mHealth applications, as well as to protect the privacy of their users. The paper claims that even though Android provides security mechanisms such as permissions ans sandboxing, mHealth applications are still plagued by serious privacy and security issues since the security model of Android is short of completely ensuring the privacy and security of users' data. A paper, supervised by a Google presents an attacker-centric threat model for mobile platforms and discusses the types of malicious attacks mobile applications can have. The threat model addresses three key issues of mobile device security: attacker's goals, attack vectors and mobile malware [10].

Since the field of mHealth applications is fairly new, there has been a few articles and papers which concentrate on the testing procedures and practices for these application. In an article by Haigh and Landwehr, proposes a model with the elements of writing code for Medical Device Software, which provides a basis for reducing the risk of malicious attacks [16]. An article in 2016, presents a guide regarding security solution for developers of mHealth apps that aims to guarantee and facilitate security measures in the development of mobile health applications by programmers unconnected to the ITC and professional health areas[31]. In 2013, OWASP released Security Testing Guidelines for Mobile Apps which iterates how the security testing for mobile applications is different from web applications in threat modelling and vulnerability analysis, and provides guidelines to test mobile security which resonates with the OWASP Mobile Top Ten Security Risks[41]. In Mobile Application Testing: A Tutorial, the authors have given a short review of the types of Mobile Application testing and encouraged to implement a reusable and cost-effective environment for testing mobile applications and an elastic infrastructure to support large-scale test automation to cope with frequent upgrades of mobile devices and technologies [12].

6 Conclusion

The emergence and growth of health and medical apps have not only changed human lives as it was, it has also impacted the decision-making from a political, social and cultural perspective, says author Lupton in her article 'Apps as Artefacts: Towards a Critical Perspective on Mobile Health and Medical Apps' [25]. The development and digitisation of health through Information technology has the capacity to change the dimensions of the current world. Mobile health technology has the potential to increase healthcare quality, expand access to services, reduce costs, and improve personal wellness and public health. But on the other hand, it also raises significant privacy and security challenges. The ongoing crisis of the pandemic shows how mobile technology can play a vi-

tal role in the future in healthcare in multiple ways. The US spends more than 22.6 percent of it's annual government expenditure on healthcare, as of 2018 [30]. This percentage has doubled over the past 30 years and is the highest of any nation in the world. Over 75 percent of these costs are due to the management of chronic diseases, which currently affect 45 percent of the US population. By 2023, the annual costs to manage chronic diseases alone are expected to rise to $4.2 trillion [11]. Similar challenges occur in many developed nations with an aging citizenry, and in developing nations that strive to provide better healthcare to their growing populations. The development of mHealth can play an important role in changing the dimensions of these costs and can enable healthcare more accessible, affordable and available. mHealth apps have become the third-most popular category of Android apps in the last few years. The invention and development of wearable devices like smart watches, ECG monitors, blood pressure monitors, etc have made things easier for the users, but have simultaneously made security and privacy a vital issue. While the apps have been improving when it comes to security and privacy in mobile apps, it is far behind than where it needs to be. There are active researches working to finding better security solutions everyday. Hence, the field of research in mHealth applications is only ever-increasing. A point worth mentioning is that since more work is being done concerning Android applications, and working with iOS applications is rarer, there is much more that can be done to ensure security and privacy in iOS applications. However, all that said, when it comes to security and privacy of mHealth applications, there's is a long way to go from where we are now.

References

[1] G. Addonizio. The privacy risks surrounding consumer health and fitness apps, associated wearable devices, and hipaa's limitations. 2017.

[2] S. Akter, J. D'Ambra, and P. Ray. Development and validation of an instrument to measure user perceived service quality of mhealth. *Information & Management*, 50(4):181–195, 2013.

[3] Y. Cifuentes, L. Beltrán, and L. Ramírez. Analysis of security vulnerabilities for mobile health applications. In *2015 Seventh International Conference on Mobile Computing and Networking (ICMCN 2015)*, 2015.

[4] P. R. Clearinghouse. Mobile health and fitness apps: What are the privacy risks. *Retrieved September*, 7:2013, 2013.

[5] I. I. Company. Kiuwan.

[6] C. Cornelius, R. Peterson, J. Skinner, R. Halter, and D. Kotz. A wearable system that knows who wears it. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 55–67, 2014.

[7] K. Crager and A. Maiti. Information leakage through mobile motion sensors: User awareness and concerns. In *Proceedings of the European Workshop on Usable Security (EuroUSEC)*, 2017.

[8] A. K. Das, P. H. Pathak, C.-N. Chuah, and P. Mohapatra. Uncovering privacy leakage in ble network traffic of wearable fitness trackers. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, pages 99–104, 2016.

[9] T. Dehling, F. Gao, S. Schneider, and A. Sunyaev. Exploring the far side of mobile health: information security and privacy of mobile health apps on ios and android. *JMIR mHealth and uHealth*, 3(1):e8, 2015.

[10] G. Delac, M. Silic, and J. Krolo. Emerging security threats for mobile platforms. In *2011 Proceedings of the 34th International Convention MIPRO*, pages 1468–1473. IEEE, 2011.

[11] R. DeVol, A. Bedroussian, A. Charuworn, A. Chatterjee, I. Kim, S. Kim, and K. Klowden. An unhealthy america: The economic burden of chronic disease. *Santa Monica, CA: Milken Institute*, 326:2010–2060, 2007.

[12] J. Gao, X. Bai, W.-T. Tsai, and T. Uehara. Mobile application testing: a tutorial. *Computer*, 47(2):46–55, 2014.

[13] V. Gay and P. Leijdekkers. Bringing health and fitness data together for connected health care: mobile apps as enablers of interoperability. *Journal of Medical Internet Research*, 17(11), 2015.

[14] Q. Grundy, F. P. Held, and L. A. Bero. Tracing the potential flow of consumer data: a network analysis of prominent health and fitness apps. *Journal of medical Internet research*, 19(6):e233, 2017.

[15] Q. H. Grundy, Z. Wang, and L. A. Bero. Challenges in assessing mobile health app quality: a systematic review of prevalent and innovative methods. *American journal of preventive medicine*, 51(6):1051–1059, 2016.

[16] T. Haigh and C. Landwehr. Building code for medical device software security. *IEEE Cybersecurity*, 2015.

[17] D. He, M. Naveed, C. A. Gunter, and K. Nahrstedt. Security concerns in android mhealth apps. In *AMIA Annual Symposium Proceedings*, volume 2014, page 645. American Medical Informatics Association, 2014.

[18] R. Herbster, S. DellaTorre, P. Druschel, and B. Bhattacharjee. Privacy capsules: Preventing information leaks by mobile apps. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pages 399–411, 2016.

[19] M. Hussain, A. Zaidan, B. Zidan, S. Iqbal, M. Ahmed, O. Albahri, and A. Albahri. Conceptual framework for the security of mobile health applications on android platform. *Telematics and Informatics*, 35(5):1335–1354, 2018.

[20] IBMSecurity and the Ponemon Institute. Ibm sponsored study finds mobile app developers not investing in security, March 2015.

[21] S. Khandelwal. Over 500 android apps on google play store found spying on 100 million users.

[22] D. Kotz, C. A. Gunter, S. Kumar, and J. P. Weiner. Privacy and security in mobile health: A research agenda. *Computer*, 29(6):22–30, 2016.

[23] D. Lai. Googleplaycrawler.

[24] LinkedIn. Quick android review kit.

[25] D. Lupton. Apps as artefacts: Towards a critical perspective on mobile health and medical apps. *Societies*, 4(4):606–622, 2014.

[26] R. Mahmood, N. Esfahani, T. Kacem, N. Mirzaei, S. Malek, and A. Stavrou. A whitebox approach for automated security testing of android applications on the cloud. In *2012 7th International Workshop on Automation of Software Test (AST)*, pages 22–28, 2012.

[27] J. Mannino. Owasp top 10 mobile risks.

[28] B. Martínez-Pérez, I. De La Torre-Díez, and M. López-Coronado. Privacy and security in mobile health apps: a review and recommendations. *Journal of medical systems*, 39(1):181, 2015.

[29] Matlink. gplaycli.

[30] O. mondiale de la santé. *World Health Statistics 2018: Monitoring Health for the SDGs Sustainable Development Goals*. World health organization., 2018.

[31] E. P. Morera, I. de la Torre Díez, B. Garcia-Zapirain, M. López-Coronado, and J. Arambarri. Security recommendations for mhealth apps: elaboration of a developer's guide. *Journal of medical systems*, 40(6):152, 2016.

[32] B. Olabenjo and D. Makaroff. Information leakage in wearable applications. In *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pages 211–224. Springer, 2019.

[33] F. Olano. google-play-api.

[34] F. Olano. google-play-scraper.

[35] A. Papageorgiou, M. Strigkos, E. Politou, E. Alepis, A. Solanas, and C. Patsakis. Security and privacy analysis of mobile health applications: the alarming state of practice. *IEEE Access*, 6:9390–9403, 2018.

[36] N. Raval, A. Razeen, A. Machanavajjhala, L. P. Cox, and A. Warfield. Permissions plugins as android apps. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 180–192, 2019.

[37] J. Sametinger, J. Rozenblit, R. Lysecky, and P. Ott. Security challenges for medical devices. *Communications of the ACM*, 58(4):74–82, 2015.

[38] B. H. Sampat and B. Prabhakar. Privacy risks and security threats in mhealth apps. *Journal of International Technology and Information Management*, 26(4):126–153, 2017.

[39] K. Scott, D. Richards, and R. Adhikari. A review and comparative analysis of security risks and safety measures of mobile health apps. *Australasian Journal of Information Systems*, 19, 2015.

[40] R. Sharp. Lacking regulation, many medical apps questionable at best. *New England Center for Investigative Reporting*, 18, 2012.

[41] F. Stahl and J. Ströher. Security testing guidelines for mobile apps. *AppSec Research EU, The OWASP Foundation [PDF File]. Retrieved October*, 18:2015, 2013.

[42] S. R. Steinhubl, E. D. Muse, and E. J. Topol. The emerging field of mobile health. *Science translational medicine*, 7(283):283rv3–283rv3, 2015.

[43] A. Sunyaev, T. Dehling, P. L. Taylor, and K. D. Mandl. Availability and quality of mobile health app privacy policies. *Journal of the American Medical Informatics Association*, 22(e1), 2015.

[44] F. Zubaydi, A. Saleh, F. Aloul, and A. Sagahyroon. Security of mobile health (mhealth) systems. In *2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 1–5. IEEE, 2015.