Journal of the American Medical Informatics Association, 00(0), 2021, 1–11
doi: 10.1093/jamia/ocab131
Research and Applications

Research and Applications

Analyzing security issues of android mobile health and medical applications

Gioacchino Tangari,[1] Muhammad Ikram,[1] I. Wayan Budi Sentana,[1] Kiran Ijaz,[2] Mohamed Ali Kaafar,[1] and Shlomo Berkovsky[2]

[1]Department of Computing, Macquarie University, Sydney, Australia [2]Centre for Health Informatics, Australian Institute of Health Innovation, Macquarie

Corresponding Author: Muhammad Ikram, Department of Computing, Macquarie University, Becton Dickinson (BD) Building, 4 Research Park Drive, Macquarie Park, Sydney, New, South Wales 2113, Australia (Muhammad.Ikram@mq.edu.au)

Received 11 February 2021; Editorial Decision 9 June 2021; Accepted 11 June 2021

ABSTRACT

Objective: We conduct a first large-scale analysis of mobile health (mHealth) apps available on Google Play with the goal of providing a comprehensive view of mHealth apps' security features and gauging the associated risks for mHealth users and their data.
Materials and Methods: We designed an app collection platform that discovered and downloaded more than 20 000 mHealth apps from the *Medical* and *Health & Fitness* categories on Google Play. We performed a suite of app code and traffic measurements to highlight a range of app security flaws: certificate security, sensitive or unnecessary permission requests, malware presence, communication security, and security-related concerns raised in user reviews.
Results: Compared to baseline non-mHealth apps, mHealth apps generally adopt more reliable signing mechanisms and request fewer dangerous permissions. However, significant fractions of mHealth apps expose users to serious security risks. Specifically, 1.8% of mHealth apps package suspicious codes (eg, trojans), 45.0% rely on unencrypted communication, and as much as 23.0% of personal data (eg, location information and passwords) is sent on unsecured traffic. An analysis of the app reviews reveals that mHealth app users are largely unaware of the surfaced security issues.
Conclusion: Despite being better aligned with security best practices than non-mHealth apps, mHealth apps are still far from ensuring robust security guarantees. App users, clinicians, technology developers, and policy makers alike should be cognizant of the uncovered security issues and weigh them carefully against the benefits of mHealth apps.

Key words: mobile health and medical application, static analysis, dynamic analysis, security, mobile malware

INTRODUCTION

With the steady growth in populations having access to smartphone devices, we have witnessed an explosion of mobile applications (in short, apps) available through various online marketplaces. As of late 2020, there were approximately 2.56 million apps[1] available on Google Play alone. Breaking these by category, we note 2 popular, mutually exclusive categories of *Medical* and *Health & Fitness* apps. Referred to collectively as mobile health (or *mHealth*) apps, these encompass a range of functions, spanning from chronic condition management and symptom checkers to step/calorie counters and period trackers.[2] Reflecting the growth of this app segment, recent guidelines of the US Food and Drug Administration formalized the use of mHealth apps for healthcare and recommended to consider those providing aid to patients or clinicians as medical devices.[3]

While the potential of mobile health to improve real-time monitoring and access to healthcare resources is well established,[4,5] mHealth apps can pose serious risks to users. Many mHealth apps offer no validation measures of effectiveness from the medical standpoint,[6] and a range of potential safety issues has been identified.[7] In addition, concerns around the security of mHealth apps are particularly topical due to the sensitive type of information such apps collect and access and the potential risks associated with data breaches or tampering.[8,9] (The dataset and analysis scripts are available at https://mhealthappsec.github.io/)

Despite the advantages over the non-mHealth baseline, mHealth apps are still far from offering robust security guarantees. Turning to communication security, we observed a considerable fraction of mHealth app communications on unsecured flows, even when transmitting sensitive user data, such as location and password. This is alarming, given the recent reports on Internet surveillance and unwanted commercialization of user data by network operators.[10,11]

The threat to patient data confidentiality and integrity, as well as the limited quality and safety improvements observed for top mHealth apps,[12] motivate the urgent need for auditing this segment of apps, evaluating their security practices and inherent flaws, and investigating the user perceptions of key aspects of security. While previous research targeted the security aspects of mHealth apps, the previously reported analyses[4,13–16] covered fewer than 100 apps and primarily capitalized on manual security tests. In this study, we embark on a large-scale security analysis of more than 20 000 mHealth apps available on the Google Play store[17] and deploy a suite of automated app collection and analysis tools. Our study covers a large number of mHealth apps and investigates a wide range of mHealth app security aspects, spanning from the verification of app sources and protection of app communication, to the presence of malicious activities in the apps and the risks of over-privileged apps. In contrast to prior studies,[4,13–16] our study takes a step forward by showing that security vulnerabilities and outdated practices in mHealth apps pose serious risks to the integrity and confidentiality of user information, and that these risks are generally unnoticed by the app users.

MATERIALS AND METHODS

Mhealth app discovery and collection

Google Play neither provides a complete list of mHealth apps nor does its search functionality yield all the available apps. To overcome this and detect as many mHealth apps as possible, we developed a crawler that interacted directly with the store's interface. Starting from the top-100 apps from the Medical and Health & Fitness categories on Google Play, the crawler systematically searched through other apps considered *similar* by Google Play (ie, other apps presented on Google Play's apps pages in the section "Similar apps" that belong to the same category). For each app, the crawler collected the following metadata: app category and price, locations where the app is available, app description, number of installs, developer information, user reviews, and app rating. For baseline comparison purposes, we repeated the same methodology to sample a set of popular non-mHealth apps from the Tools, Communication, Personality, and Productivity categories on Google Play.

Analysis methodology

Guided by security and privacy analyses of Android apps outlined in our previous work,[18–20] we depicted our security analysis of mHealth apps in Figure 1. To provide a comprehensive view of mHealth apps' security, we combined the analysis of app resources and source code (known as *static analysis*) with the analysis of app runtime behavior (known as *dynamic analysis*). In addition, we analyzed all public app reviews to shed light on the users' perceptions of mHealth app security.

Static app analysis

We downloaded the apps in the Android Application Package (APK) format. To analyze the apps, we decompiled the APKs using APKTool (APKTool is a tool for reverse engineering Android apps: https://ibotpeaches.github.io/Apktool/), allowing us to decode the compiled sources to their *nearly* original form.[21] We briefly explain how we analyzed the extracted app resources:

- *App signing security.* All Android APKs are digitally signed with a certificate before they are installed or updated. App developers use a cryptographic signature (signing key) verifying the package as legitimate. The signing process attaches a certificate to an app, associating the APK to the developer's identity and private key. Hence, the security of the signing process entirely relies on the secrecy of the signing key.
- *Encryption schemes* like MD5 for certificate signing are considered weak and insecure[22] compared to other methods (eg, SHA-256 hashing). If this becomes publicly available, anyone can use it to sign a potentially malicious app that claims to be an update to a legitimate app, and users' phones will update the real app with the malicious one. We analyzed the app certificates (obtained from APSuch difference between mHealth and non-mHealth apps in seKTool's output) using a custom-built script based on keytool.[23] For each app, we measured the encryption scheme used for signing the app certificate and the key length.
- *Malware presence.* To identify suspicious mHealth apps, we inspected the APKs using the VirusTotal[24] public API, which aggregates the scanning capabilities of 68 popular antivirus tools. For each analyzed app, we obtained *malware label* and number of *malware positives* showing, respectively, the class of malware, such as trojan or adware, and the agreement among the antivirus tools in classifying the app as malware. VirusTotal has been commonly used to detect malicious apps, executables, software, and domains.[18,19,25–27] As each of VirusTotal's antivirus tools may produce false positives, we computed the aggregate *AVScore* metric (ie, the number of tools that flagged an APK as malicious) with maximal score being 68. To minimize the occurrence of false positives and obtain a clear indication of malicious activity, we restricted our further analyses to apps having $AVScore \geq 5$, in agreement with previous studies on Android app malware.[19,28]
- *Apps' permissions.* Android apps request *permissions* for accessing system resources, such as contact list and camera. The type and number of the requested permissions are indicative of *potential* attack surfaces[29] as well as security vulnerabilities in Android apps.[30] To analyze mHealth permissions, we *first* obtained the requested apps' permissions from the app manifest files (AndroidManifest.xml) by parsing the *uses-permission* and *service* tags. We then measured the frequency of permission requests by mHealth apps, number of permissions per app, and presence of sensitive permissions requests. For each permission we deployed the PScout[31] API/permission mapping to check if the app code contains at least 1 API call requiring that permission. For sensitive permission analysis, we leveraged Android docu-

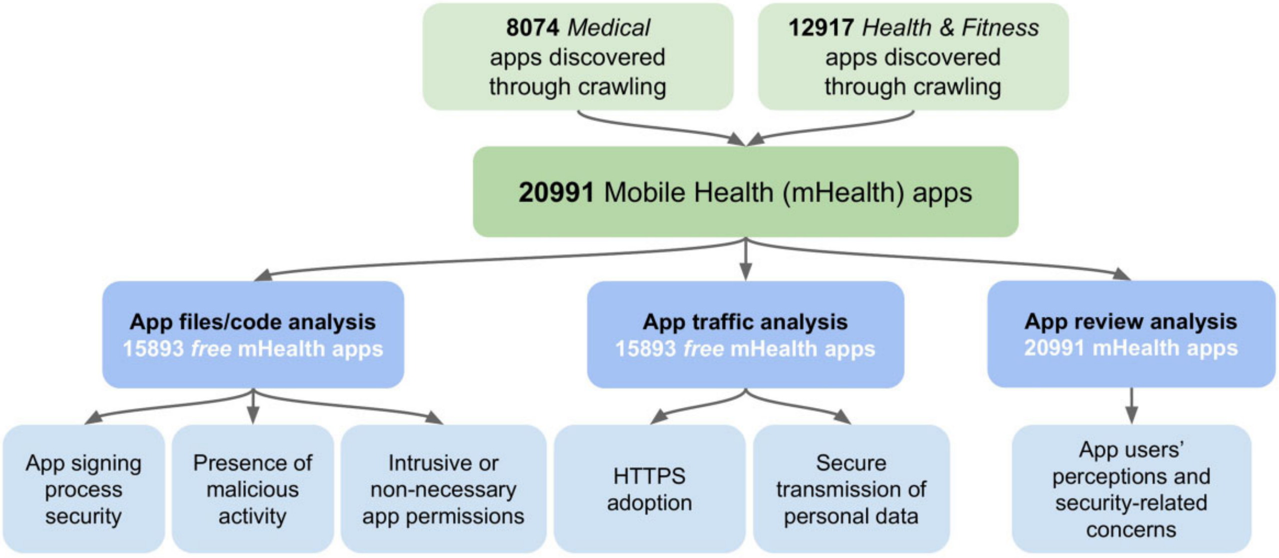
8074 Medical apps discovered through crawling
12917 Health & Fitness apps discovered through crawling
20991 Mobile Health (mHealth) apps
App files/code analysis
15893 free mHealth apps
App traffic analysis
15893 free mHealth apps
App review analysis
20991 mHealth apps
App signing process security
Presence of malicious activity
Intrusive or non-necessary app permissions
HTTPS adoption
Secure transmission of personal data
App users' perceptions and security-related concerns

Figure 1. Overview of the mHealth app security analysis.

mentation[32] that defines *dangerous* permissions as highly sensitive.

Dynamic app analysis

A gold standard for identifying privacy leaks is to manually log into the apps and interact with them. However, this approach turns out impractical at scale. To automate the analysis, we rely on Android's UI/Application Exerciser Monkey Runner,[8] a command-line tool that generates pseudo-random user activities such as opening an app, clicking on buttons, etc. While running an app, we use *mitm-proxy*[33] (a TLS-capable interception proxy) to capture all app-generated traffic (HTTP and HTTPS) on a dual-stack[34] WiFi in our testbed. Prior work showed that synthetic usage patterns could lead to underestimating the number of privacy leaks compared to manual (human) interactions,[35] since random streams of Monkey Runner events may overlook some function calls. While this is a common drawback of automation approaches, Android's Monkey Runner exhibits the best code coverage among the existing automation tools.[36]

To execute app activities (eg, open the app, open the menu, click on buttons, etc), we *first* parsed AndroidManifest.xml file of each app, and then deployed Monkey Runner to execute all the activities of the app (*on average*, we execute 35 activities per analyzed app with a 180-second test session). To reduce traffic contamination, we minimized as much as possible all the background processes of the smartphone (eg, notifications of other apps). The traffic interception provided a detailed list of HTTP and HTTPS communications of each analyzed app.

- *Secure communication adoption.* To further assess the vulnerability of mHealth apps, we analyzed the security of their data transmission by measuring the fraction of communications on HTTP and HTTPS protocols. While HTTP uses an unencrypted plain-text format, HTTPS adopts message encryption through the SSL/TLS protocols, thus protecting users from malicious data interception and content tampering. We analyzed the app traffic captured in our testbed by measuring the number of requests adopting HTTP and HTTPS.

App review analysis

By analysing the mHealth app reviews, we quantified the users' subjective perceptions of the apps. In particular, we focus on the 1-, 2-, and 3-star reviews of apps with average ratings below 3 to investigate the concerns around the app functionality and security conduct. We obtained the complete list of reviews from the app's home page on the Google Play store. In previous work,[18,19] we used an automated keyword-based search method (we curated a list of 59 keywords mapped to 12 complaint categories [provided in Supplementary Appendix A]; for example, the keyword "crash" is mapped to the category "bugs," while "spyware" is mapped to "security") and employed manual classification of the review text to identify the key types of complaints in the reviews. First, we created dictionaries of keywords (see Supplementary Table 5), belonging to various complaint categories to filter reviews and then performed manual validation of the resultant complaints' categories. Three coauthors were involved in manual validation. Based on this manual observation, each author reclassified each user comment into 1 of the 6 classes of complaints listed in Table 4: malware, permission, security, bugs, battery, and mobile data. We used . majority voting to resolve disagreements among the coauthors. If consensus was not reached, the user's comment was marked as unlabeled and discarded from the analysis.

RESULTS

Overall, we discovered 20 991 (filtered from approximately 1.7 million apps crawled from October 1 to November 15, 2019; we then selected apps belonging to the Medical and Health & Fitness categories on Google Play) mHealth apps, of which 15 893 (75.7%), 3228 (15.4%), and 1872 (8.9%) belonged to the free, paid, and geo-blocked (could not be downloaded from Australia) categories, respectively. Table 1 summarizes our dataset. For baseline comparison, we collected 8468 popular non-mHealth apps belonging to the Tools, Communication, Personality, and Productivity categories. We argue this constitutes a reasonable sample of non-mHealth apps considered for comparative analyses.

Table 1. Summary of the 20 991 mHealth apps collected from the Google Play store

Characteristics	No. (%)
0%–20%	10 371 (49.4%)
mHealth category	20 991 (100%)
Medical	8074 (38.5%)
Health & Fitness	12 917 (61.5%)
Fee required to download	
Yes (paid mHealth apps)	3228 (15.4%)
No (free mHealth apps)	15 893 (75.7%)
No (geoblocked mHealth apps)	1872 (8.9%)
# of Downloads	
500+	7481 (35.9%)
1000+	4009 (19.2%)
5000+	1683 (8.1%)
10 000+	3582 (17.2%)
50 000+	1253 (6.0%)
100 000+	1882 (9.0%)
500 000+	375 (1.8%)
1 000 000+	462 (2.2%)
5 000 000+	127 (0.6%)
Avg. Rating	
0.0–1.0	6146 (29.3%)
1.0–2.0	240 (1.1%)
2.0–3.0	1350 (6.4%)
3.0–4.0	4856 (23.1%)
4.0–5.0	8396 (40.0%)
User perception determined by 100% × ofnegativereviews/ofall reviews	
20%–40%	4157 (19.8%)
40%–60%	2663 (12.6%)
60%–80%	1474 (7.0%)
80%–100%	2326 (11.1%)

Certificate analysis

To evaluate the security of the app key-signing process, we analyzed the encryption scheme deployed for signing the app certificate and the length of the public keys employed in the signing process. Table 2 reports our results. We observed that the vast majority of mHealth apps used MD5, SHA256, and SHA1 hashing with RSA encryption (<1%, 56.6%, and 38.9%, respectively), with minor differences between Medical and Health & Fitness apps. Compared to non-mHealth apps, we noted that mHealth apps were better aligned with new signing schemes as shown by the higher fraction of apps using SHA-256 (56.6% vs 36.2%) (x^2 statistic = 943.0, P value < .001) and a lower fraction using weak algorithms (in particular, < 1% vs. 3.4% for MD5) (x^2 statistic = 251.5, P value < .001).

In addition to the signing scheme, we measured the length of the public keys. Generally speaking, increasing the key length complicates cracking the signing key, which strengthens the app security. The current standard (eg, for SSL certificates) is 2048 bits, with 1024-bits keys being considered obsolete. Table 2 shows that a higher fraction of mHealth apps met the current standard than non-mHealth apps: 57.8% used 2048-bits key vs 45.2% for non-mHealth apps (x^2 statistic = 390.9, P value < .001). At the same time, 10.4% of mHealth apps were already using 4096-bits keys, which is meant to be the next step for RSA keys, compared to 3.9% for non-mHealth apps (x^2 statistic = 317.1, P value < .001).

Malware and trojan presence

Upon inspecting the code and files of mHealth apps using VirusTotal, we found 378 apps (1.8% of apps) being labeled as suspicious by at least 1 antivirus tool. As shown in Table 3, VirusTotal reports malicious activity for 257 (2.0%) Health & Fitness apps and 121 (1.5%) Medical apps. These fractions are smaller than those found in the non-mHealth app set, where VirusTotal flagged 616 apps out of 8468 (7.3%). Table 3 also shows that approximately 70% of the suspicious mHealth apps (264 out of 378) were only flagged by 1 antivirus tool. To minimize the occurrence of false positives and obtain a clear indication of malicious activity, we restricted our analyses to apps having AVScore $\geq$ 5, in agreement with previous studies on Android app malware.[19,28] This set includes 53 apps: 17 Medical and 36 Health & Fitness. Table 3 reports the top-10 apps according to AVScore.

Although this set is relatively small, we noted that some Health & Fitness apps with the highest AVScores had been downloaded between 100 000 and 1 million times (eg, *Aapa k Totkay*[37] and *Health Benefit*).[38] Others, such as *Nursing Care*[39] or *Infection Prevention*,[40] had average ratings above 4.5 despite having AVScores of 19 and 11, respectively. To further investigate the 53 malicious mHealth apps, we employed the AVClass[41] and Euphony[42] tools, allowing us to determine the malware type/family from the antivirus scan labels. Following the AVScore $\geq$ 5 criterion and the results of AVClass and Euphony, we found 41 apps with Adwares (adware is advertising-supported software that helps to increase the revenues of other software. The ads may be within the software itself or may encourage installation of additional software by third-party sponsors. Most adware is safe and legitimate, while some conduct malicious activity.[43]), 10 apps with trojans (a trojan is a malware often disguised as legitimate software. Users are lured by some form of social engineering into loading and executing trojans. Once activated, trojans allow cyber-criminals to steal sensitive data or gain backdoor system access.[44]), and 2 apps containing riskware and other undefined malware. Below, we focus on the 2 main malware types in mHealth apps: adware and trojans.

Adware presence

We identified several instances of well-known adware families: Airpush (14 samples), Leadbolt (8 samples), and Revmob (5 samples). These are originally legitimate mobile advertisement libraries commonly used by developers to monetize their apps; however, their inclusion in the apps poses 2 important security issues. On several occasions these adware were exploited as vehicles of malware payloads. Multiple reports[17,45,46] revealed that Airpush was exploited to download malware and trojans. In these cases, the malicious code was nested within Airpush to evade virus detection, as the app was only labeled as adware. Moreover, several domains associated with these services (eg, *ad.leadboltapps.net* and *au.umeng.com*) were identified as malicious in previous analyses.[47] In addition, these services aggressively collect personal user information, including hardware/device identifiers and location data. In the traffic analysis reported below, we found evidence of the 3 aforementioned adware collecting the sensitive IMEI and MAC hardware identifiers, as well as GPS locations.

Trojan presence

Our analysis discovered 3 samples of the FakeApp trojan family. FakeApp trojans often masquerade as part of a legitimate app—in many cases, an antivirus app or an updater.[48] They then try to gain revenues by displaying intrusive ads or redirecting users to installing third-party apps. Several variants of this trojan have been reported to perform malicious activities, such as harvesting user credentials

Table 2. Top: analysis of app signing schemes; bottom: analysis of public key length. The results are broken down into the Medical and Health & Fitness categories. We also included the results for the baseline non-mHealth apps

Signing scheme	All mHealth apps	Health & fitness apps	Medical apps	non-mHealth apps
SHA256 + RSA Encryption	11 884 (56.6%)	7480 (57.9%)	4361 (54.0%)	3061 (36.2%)
SHA1 + RSA Encryption	8172 (38.9%)	4747 (36.8%)	3494 (43.3%)	4960 (58.6%)
DSA + SHA1	761 (3.6%)	601 (4.6%)	127 (1.6%)	285 (3.4%)
MD5 + RSA Encryption	172 (< 1%)	88 (< 1%)	91 (1.1%)	1 (< 1%)
SHA512 + RSA Encryption	2 < 1%	2 (< 1%)	2 (< 1%)	2 (< 1%)
SHA1 + RSA	–	–	–	–
Key length (bits)	All mHealth apps	Health & fitness apps	Medical apps	non-mHealth apps
2048	12 143 (57.8%)	7401 (57.3%)	4760 (59.0%)	3825 (45.2%)
1024	6660 (31.7%)	4091 (31.7%)	2571 (31.9%)	4306 (50.9%)
4096	2184 (10.4%)	1423 (11.0%)	741 (9.2%)	334 (4.0%)
3072	2 (< 1%)	1 (< 1%)	0 (0.0%)	0 (0.0%)
2047	2 (< 1%)	1 (< 1%)	0 (0.0%)	2 (< 1%)
1048	2 (< 1%)	0 (0.0%)	2 (< 1%)	0 (0.0%)

and personal data.[49] The scanning also revealed 1 instance of the Trojan.Gen.2 trojan, responsible for unwanted browser redirects, advertisements, and even malicious access to credentials.[50]

Permission analysis

Frequency of requested permissions

We identified the permissions frequently requested by the different app categories. The results in Figure 2 show that the most frequent permissions were for accessing the device storage (WRITE_EXTERNAL_STORAGE and READ_EXTERNAL_STORAGE), requesting network functionality (ACCESS_WIFI_STATE), and obtaining location (ACCESS_FINE_LOCATION and ACCESS_COARSE_LOCATION). Turning to the overall app permission requests, we noted that the Medical and Health & Fitness apps behaved similarly, and both were less demanding than non-mHealth apps.

Restricting the analysis to the suspicious mHealth apps from Table 3, we observed that these tended to incorporate more sensitive permissions than the generic non-mHealth apps. For instance, 213 out of 378 (56.3%) of the suspicious apps requested the GET_ACCOUNTS permission, while the same was requested by less than 20% of the total 20 991 mHealth apps (x^2 statistic = 293.02, P value < .001).

Over-privileged apps

We also measured the number of permissions requested by each app. High numbers of permissions can indicate the apps gaining more permissions than required for their operation. Such cases, which are particularly frequent for apps preinstalled in vendor-

Table 3. Analysis of *suspicious* apps according to VirusTotal reports. Ranked list of top-10 most *suspicious* apps with their average ratings (ARating) and number of installs

App category	No. (%) apps flagged by VirusTotal)	x^2 statistic	P value
mHealth apps	378 (1.8%)	552.8	< .001
non-mHealth apps	616 (7.3%)		

AVScore	All mHealth apps (N = 378)	Health & fitness apps (N = 257)	Medical apps (N = 121)	non-mHealth apps (N = 616)
1	264 (69.8%)	176 (68.5%)	88 (72.7%)	420 (68.2%)
2	48 (12.7%)	33 (12.8%)	15 (12.4%)	56 (9.0%)
3	8 (2.1%)	7 (2.7%)	1 (< 1%)	25 (4.0%)
4	5 (1.3%)	5 (2.0%)	0 (0.0%)	20 (3.2%)
5	6 (1.6%)	5 (2.0%)	1 (< 1%)	18 (2.9%)
≥6	47 (12.4%)	31 (12.0%)	16 (13.2%)	77 (12.5%)

Health & fitness apps					Medical apps				
#	App	AV Score	Installs	ARating	#	App	AV Score	Installs	ARating
1	Urdu Best Totkays[63]	24	10 000+	3.8	1	Nursing Care[39]	19	1000+	5
2	Aapa k Totkay[37]	19	100 000+	3.8	2	FarmAlicante[64]	16	1000+	4.3
3	Yoga For Diabetes[65]	18	10 000+	3.3	3	TritionRx[66]	15	1000+	3.6
4	Guardian Angel[67]	16	5000+	4	4	Smoke'n Vap'z[68]	15	10+	n.d.
5	Fit Bites[69]	15	10000+	4.1	5	COMM[70]	14	100+	n.d.
6	iCom[71]	15	1000+	2.7	6	OptiKoncept[72]	14	10+	n.d.
7	Cellu hit[73]	15	10+	n.d.	7	Infection Prevention[40]	11	5000+	4.7
8	Your Angels[74]	15	50 000+	4.1	8	Vap'Pause[75]	11	100+	5
9	Health Benefit[38]	14	100 000+	4.3	9	Doctor Street[76]	11	5000+	3.6
10	Esthetic Medicare[77]	14	10+	n.d.	10	Fragerstrom's TTest[78]	10	500+	3.4

Table 4. User complaints in mHealth app reviews are broken down into 2 mutually exclusive categories: Medical and Health & Fitness

Complaint	All mHealth (391 642 complaints)				Medical (67 057 complaints)				Health & Fit. (324 585 complaints)			
Category	#Compl.	%Compl.	#Apps	%Apps	#Compl.	%Compl.	#Apps	%Apps	#Compl.	%Compl.	#Apps	%Apps
Usability:												
Bugs	201 240	51.4	2240	10.7	34 728	51.8	627	7.7	166 512	51.3	1613	12.5
Battery	7710	2.0	568	2.7	4784	7.1	120	1.5	2926	< 1	448	3.5
Mobile data	2058	< 1	427	2.0	169	< 1	70	< 1	1787	< 1	305	2.4
Security:												
Malware	6562	1.7	234	1.1	4132	6.2	47	< 1	2430	< 1	187	1.4
Security	4750	1.2	245	1.1	1725	2.6	72	< 1	3025	< 1	173	1.3
Permissions	7451	1.9	424	2.0	1600	2.4	86	1.0	5851	1.8	338	2.6

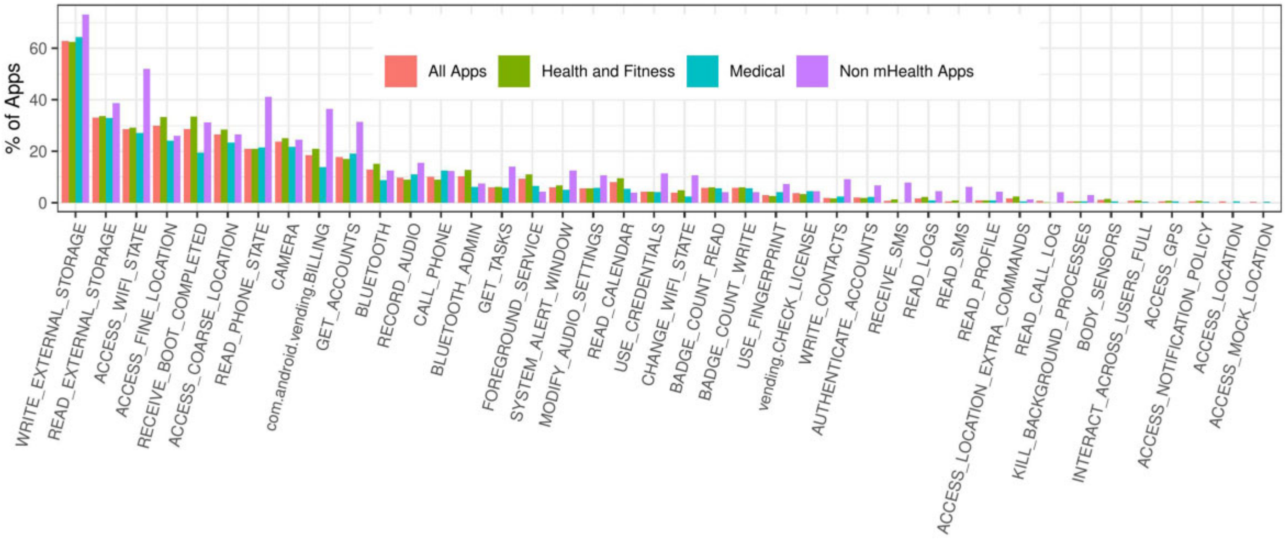

Figure 2. Permission distribution for the mHealth app categories and non-mHealth apps.

customized phones,[51] violate the principle of *least privilege* and can increase the attack surface or amplify the effect of app vulnerabilities. The results, depicted in Figure 3, showed that mHealth apps requested fewer permissions than non-mHealth apps (Figure 3a), while we observed a similar behavior across the 2 categories of mHealth apps (Figure 3b). Figure 3 also highlighted that a fraction of mHealth apps (1043 out of 20 991, corresponding to the 5.0% of the mHealth app set) requested a large number of permissions (ie, 30 or more). Focusing on the top-5% of apps with most permission requests, we investigated if such apps needed all the requested permissions. We discovered that out of the 754 mHealth apps in this top-5% set, 656 apps (87.0%) contained at least 1 non-necessary permission—we denote these apps as *over-privileged*. Notably, among the over-privileged mHealth apps requesting unnecessary permissions, we found popular Health & Fitness applications, such as *Huawei Health*[52] (100M+ downloads), where we detected 25 unused permissions out of 102, and *MyWellness*[53] (1M+ downloads), with 30 unused permissions out of the 47 requested.

Sensitive permissions

Restricting the results to dangerous permissions, defined as highly sensitive in Android documentation,[32] we observed in Figure 3 similar trends across mHealth and non-mHealth apps and the 2 mHealth app categories. We also found that the *suspicious* mHealth apps from Table 3 tended to incorporate more sensitive permissions than non-mHealth apps. For instance, 36% of suspicious apps requested the READ_CONTACTS permission, which was rarely present across other mHealth apps. This permission is particularly *sensitive*, as app developers may misuse Android's logging capabilities and inadvertently expose personal information to other apps.

Secure communication adoption

We explored the prevalence of HTTP/HTTPS in individual apps, focusing on the fraction of HTTPS requests (Figure 4a) and bytes of traffic (Figure 4b). Overall, we obtained 190 423 requests, of which 112 942 (59.3%) used HTTPS and 77 481 (40.7%) used plain-text HTTP. Based on Figure 4a, almost 35.0% of mHealth apps exclusively relied on HTTPS, while 45.0% only used HTTP, with the remaining 20.0% using both protocols. Comparing mHealth with non-mHealth apps, we observed that the former relied more on secure data transmission, as shown by the smaller fraction of HTTP requests and traffic. However, the fraction of mHealth apps that exclusively communicated in plain-text HTTP was still considerable, as 10 285 out of 20 991 mHealth apps (49.0%) relied on HTTP requests only.

To better understand the temporal trends, Figure 4c shows the use of HTTPS (measured as the average fraction of HTTPS requests) in mHealth apps with different release or update dates. We observed a steadily increasing trend in the use of HTTPS until 2017, consistent across the 2 categories of apps. From 2017, however, the

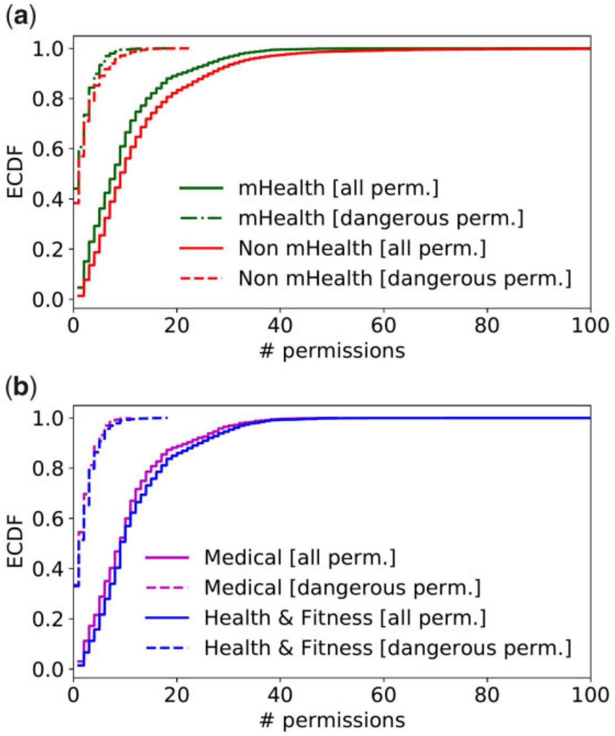

Figure 3. Empirical cumulative distribution function (ECDF) of the number of permissions requested by mHealth and non-mHealth apps.

HTTPS ratio has dropped for both the Medical and Health & Fitness apps. We conjecture that this observation could be explained by the effect of mHealth apps embedding an increasing number of multimedia resources, such as images, which are loaded over HTTP.

We also assessed the adoption of HTTPS by mHealth apps when transmitting personal (user and device) information. For this evaluation, we detected the personal data transmitted by mHealth apps by inspecting the HTTP/HTTPS headers (URI, Referrer and post Data, based on the method proposed by Ren et al[54]) and then computed the portion of transmissions on HTTP and HTTPS. The results in Figure 5 showed that as much as 23.0% of personal data was transmitted on unencrypted HTTP flows. More alarmingly, even for sensitive data, such as GPS location and password, a significant portion of transmissions (45.0%) used HTTP.

User perceptions

We obtained the complete list of reviews from the app's home page on the Google Play store. Upon excluding reviews with no text (reviews with no text include emojis such as smiley face and thumbs up) we obtained 2 130 684 reviews for 6938 mHealth apps (1 764 486 for Health & Fitness apps and 366 198 for Medical ones). We obtained 1 788 463 (83.9%) 4- and 5-star reviews and 235 210 (11.0%) 1-, 2-, and 3-star reviews. While most mHealth apps predominantly received 4- or 5-star reviews, 2324 apps (11.1%) received at most a 3-star review average. We investigated the relationship between these 1-, 2-, and 3-star reviews and the app's security conduct. Table 1 groups mHealth apps based on the fraction of 4- and 5-star reviews. A scan of the 235 210 negative reviews yielded a total of 391 642 user complaints: 67 057 for Medical apps and 324 585 for Health & Fitness apps.

Overall perceptions

In Table 4, we detailed 6 user complaint categories (note that the coauthors disagreed on < 1% of the cases, which was resolved using majority voting) of which 3 referred to the app usability, and 3 to the security conduct. Notably, most complaints (53.8%) pointed to *usability* flaws, especially to bugs (51.4%). The user complaints related to security were much less frequent: only 1.2% of complaints explicitly mentioned security, suggesting that mHealth app users had a limited interest in (or awareness of) security issues.

Security-related user complaints

A non-negligible portion (ie, 7451/391 642 = 1.9%) of negative reviews reported intrusive permissions or sensitive permission requests, an issue raised for 2.0% of apps. Explicit mentions of malware or suspicious activity ("Malware" category) were rare and appeared for only 1.1% of mHealth apps. Crucially, we discovered that only a small portion of the security issues unveiled by our analyses were actually reflected in the reviews. Firstly, no over-privileged apps identified in Section 3.3 were criticized for excessive permission requests. Secondly, considering the 46 mHealth apps with AVScore ≥ 5, likely indicating malicious or suspicious activity, only 6 apps received security or malware complaints. Thirdly, only 1 negative review explicitly mentioned the lack of secure HTTPS, whereas Section 3.4 discovered that more than 45.0% of apps relied on HTTP communications. Overall, these cases showed a significant misalignment between user perceptions of mHealth security and the observed security conduct of the apps.

DISCUSSION

While the potential of mHealth to improve real-time monitoring and access to healthcare resources is well established,[4,5] concerns around the security of mHealth apps are particularly topical due to the sensitive types of information such apps collect and access and the potential risks associated with data breaches or tampering (the dataset and analysis scripts are available at https://mhealthappsec.github.io/).[8,9] Our study has provided a detailed and comprehensive overview of security features used by mHealth apps. In the following, we summarize our findings, compare our work with previous studies, and highlight the limitations of our methodology.

Summary of findings

In general, security features offered by mHealth apps are less vulnerable than the baseline non-mHealth apps, as well as better aligned with security best practices, as mHealth apps adopt more secure signing mechanisms and more robust encryption keys. Also, by requesting slightly fewer dangerous permissions, mhealth apps may be less vulnerable to being compromised by nefarious actors.

Despite these advantages over the non-mHealth baseline, mHealth apps are still far from offering robust security guarantees. Turning to communication security, we observed a considerable fraction of mHealth app communication on unsecured flows, even when transmitting sensitive user data, such as location and password. Less than 1% of mHealth apps still used MD5, a weak certificate signing scheme. This is alarming, given the recent reports on Internet surveillance and unwanted commercialization of user data by network operators.[10,11] In addition, we found a considerable number of over-privileged mHealth apps, requesting more permissions than needed, which may compromise the integrity and confidentiality of user data in case of attacks. Our study also identified

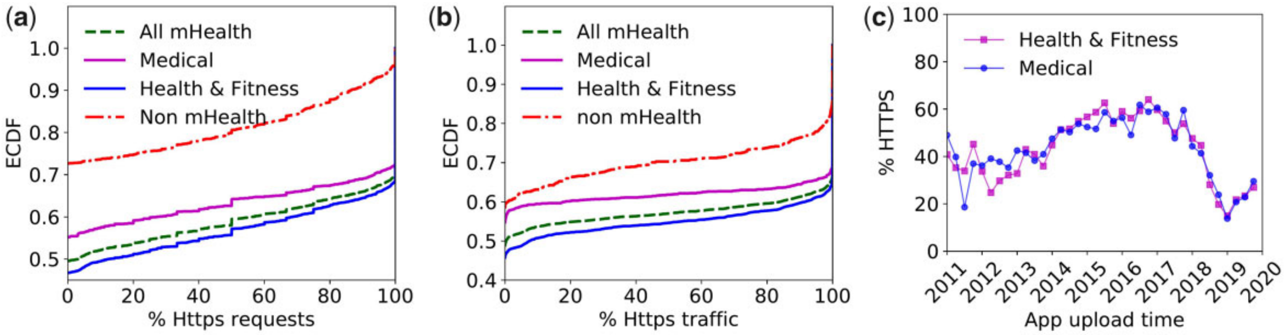

Figure 4. Adoption of the HTTPS communication protocol.

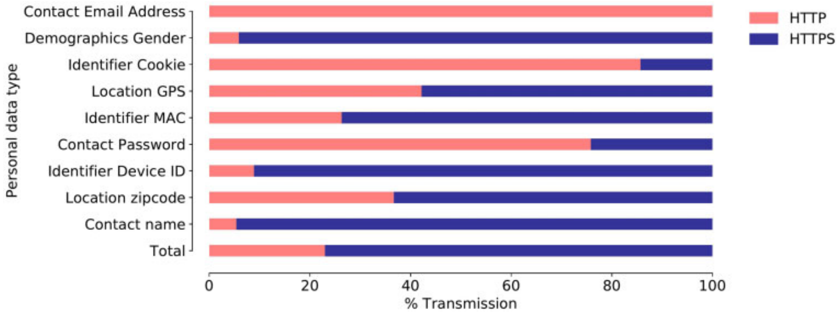

Figure 5. Transmission of personal (user or user device) information on HTTP and HTTPS traffic flows by mHealth apps.

378 (1.8%) mHealth apps incorporating malicious code or potential security threats, including occurrences of trojans and potentially malicious adware.

Although these issues are topical and potentially dangerous, our analysis of mHealth app user reviews uncovered that the users have a limited awareness of (or interest in) the security conduct of the analyzed apps. Namely, issues related to traffic security, intrusive permissions, and malicious activity presence were rarely mentioned in the reviews and were substantially less evident than performance-related issues.

Comparison with prior works

mHealth apps and the associated risks to user data have received considerable attention. Prior studies revealed a range of data collection practices carried out by health-related apps, data recipients, and their compliance with privacy policies. Huckvale et al investigated 79 health and wellness apps and found that 78% of apps transmitting user data did not describe their data collection practices in the privacy policies.[16] Blenner et al, analyzed 24 diabetes apps and discovered that 79.0% of them shared user data despite not providing any privacy policy.[13] Upon assessing the privacy practices of 36 top-ranked smoking cessation and depression apps, Huckvale et al revealed that only 12 disclosed the transmission of data to Facebook or Google in their privacy policies.[4] Grundy et al analyzed the recipients of user data collected by 24 medical apps.[14] While addressing data-collection practices and user data recipients, these works have a limited focus on the exposure of mHealth apps to security threats and their malicious behavior. Papageorgiou et al,[55] assessed the security of 20 mHealth apps using static code and dynamic traffic analysis while, by analyzing the requested permissions, He et al,[15] investigated the security vulnerabilities in 160 mHealth apps.

Our work advances prior research with respect to 2 aspects. First, we increase by several orders of magnitude the breadth of the analysis: from tens-to-hundreds of apps assessed in previous studies[4,13,14,16,62] to over 20 000 apps in our analysis. To the best of our knowledge, the only study covering a comparable number of mHealth apps was performed by Dehling et al[56] that categorized mHealth apps into 3 classes of potential privacy risks. Previous studies mainly used manual or semi-automated analyses to uncover security issues in a small set of mHealth apps. In comparison, our study deployed automated tests identifying security vulnerabilities and malicious activities in the apps' code, as well as in network traffic.

Improving mHealth security

Our study further confirms that mHealth apps have considerable security vulnerabilities. Issues like the adoption of weak encryption schemes (SHA1, SHA256, and RSA) show that app developers may not keep up with new protection standards. At the same time, although reasonable for monetization purposes, the incorporation of suspicious third-party libraries (eg, adware) may result in cloning or repackaging of mHealth apps with malicious code. Although Google uses Bouncer[57] to analyze vulnerable and malicious code in apps, this can be fingerprinted and evaded[58–60] by sophisticated obfuscation techniques. To improve mHealth security, it is imperative for app marketplaces like Google Play to devise and enforce a comprehensive security vetting policy, not only to detect vulnerable and suspicious code, but also to encourage developers to adopt stronger defenses against potential app repackaging and cloning.

Similarly, several adjustments are required from app developers to better mitigate mHealth security issues: use of fresh (and valid) certificates with robust keys, adoption of certificate pinning to minimize traffic interception, use of anticloning techniques to avoid malware code injection, and full adoption of the HTTPS protocol for protecting user data confidentiality and integrity. Finally, to contribute to securing mHealth apps and protecting sensitive data, app users (eg, patients, clinicians, and doctors) should pay attention to the requested permissions, app ratings, and user reviews before installing mHealth apps.

Limitations

To scale up the study and include a large number of apps, we relied on automated analysis methods with coverage limitations. Specifically, live testing of the apps extensively deployed randomized interactions, as opposed to hand-crafted app usage patterns, with the drawback of some components of the apps (eg, tabs, views, menus) potentially not being triggered during testing. We mitigated this issue by adopting a large app interaction window based on the apps' automation approach proposed by Ren et al,[54,61] and employed a testing automation tool providing the best app activity *coverage* among the existing solutions. Compared to Ren et al,[54,61] which analyzed privacy and usability of only 512 unique application, we provide the first comprehensive security analysis of 20 991 unique apps on Google Play store.

In addition to the automation issues, it should be highlighted that we restricted most analyses to free apps. We conjecture this does not diminish the generalizability of our findings, as only 15.4% of mHealth apps on Google Play were paid. Moreover, since VirusTotal has a limit on the scanned file size, we were unable to inspect 2236 out of the 15 852 downloaded APKs (14.1%).

CONCLUSION

mHealth apps have recently emerged as sources of health information, monitoring tools for patients, and offering decision-support for clinicians. Our large-scale analysis unveiled alarming security issues and presence of malware codes in mHealth apps, of which app users have very limited awareness. We argue that it is critical to bring these findings to the attention of the app users, clinicians, technology developers, and policy makers alike. They all should become cognizant of the uncovered issues and weigh them carefully against the benefits offered by the apps. Overall, the security issues highlighted in this article and the lower fraction of users' complaints suggest that mHealth users are unaware of the security and privacy risks of mHealth apps.

Overall, this article calls for a thorough technical and policy discussion around balancing the benefits of mHealth apps with the security and privacy risks they pose. App users, clinicians, technology developers, and policy makers alike should be cognizant of the uncovered security issues and weigh them carefully against the benefits of mHealth apps. Our study warrants community (clinicians and end users) awareness through better security training. Platform providers, such as Google Play, should employ novel tools and techniques to better inform users about potential security issues. Besides the need for medical practitioners to familiarize themselves with the privacy aspects of mHealth apps, we believe it is imperative for mobile app marketplaces, such as Google Play, to thoroughly examine the app privacy statements prior to making the apps available. The marketplaces should also ensure that their app-vetting process is up to date and able to deal with newer, sophisticated malware codes – unlike the current situation, where we observed that 1.8% of the mHealth apps are detected as malware by at least 1 antivirus tool.

FUNDING

This work was funded by Optus Macquarie University Cyber Security Hub (OMCSH); the research was also supported by the National Health and Medical Research Council (NHMRC) grant APP1134919. GT and KI were supported by postdoctoral fellowships of Macquarie University. OMCSH and the NHMRC Centre of Research Excellence (CRE) in Digital Health had no role in the study design; in the collection, analysis, and interpretation of data; in the writing of the article; or in the decision to submit the article for publication.

AUTHOR CONTRIBUTIONS

GT participated in the design of the study, led the data analysis, and wrote the first draft of the manuscript. MI secured funding, participated in the design of the study and the analysis, and led the data collection. IW participated in the app malware analysis. KI participated in the data collection and the analysis of user reviews. DK participated in the design of the project and funding acquisition. SB participated in the design of the study and acquired funding. All the authors critically revised the manuscript drafts and approved the submission.

SUPPLEMENTARY MATERIAL

Supplementary material is available at *Journal of the American Medical Informatics Association* online.

ETHICAL APPROVAL

Ethical approval was not sought as the study did not involve human subjects .

DATA AVAILABILITY STATEMENT

A sample of our dataset is available at https://mhealthappsec.github.io/. Upon publication, we will release our dataset and analysis scripts for further research.

CONFLICT OF INTEREST STATEMENT

None declared.

REFERENCES

1. Iqbal M. App download and usage statistics (2020). https://www.businessofapps.com/data/app-statistics/, 30/10/2020 Accessed February 8, 2021.
2. Kay M, Santos J, Takane. *mHealth: New Horizons for Health through Mobile Technologies*, vol 64, Geneva, Switzerland: World Health Organization; 2011, pp 66–71.
3. FDA. Digital health criteria. https://www.fda.gov/medical-devices/digital-health/digital-health-criteria (Accessed March 23, 2018).
4. Huckvale K, Torous J, Larsen ME. Assessment of the data sharing and privacy practices of smartphone apps for depression and smoking cessation. *JAMA Network Open* 2019; 2 (4): e192542.
5. Tighe J, Shand F, Ridani R, Mackinnon A, Mata NDL, Christensen H. Ibobbly mobile health intervention for suicide prevention in Australian indigenous youth: a pilot randomised controlled trial. *BMJ Open* 2017; 7 (1): e013518.
6. Rowland SP, Fitzgerald JE, Holme T, John P, MacGregor A. What is the clinical value of mHealth for patients? *NPJ Digit Med* 2020; 3 (1): 4.

7. Akbar S, Coiera E, Magrabi F. Safety concerns with consumer-facing mobile health applications and their consequences: a scoping review. *J Am Med Inform Assoc* 2020; 27 (2): 330–40.
8. monkeyrunner | Android Developers. 2019. https://developer.android.com/studio/test/monkeyrunner/ (Accessed February 5, 2021).
9. Verizon. Mobile security index. 2020. https://enterprise.verizon.com/resources/reports/mobile-security-index/#report (Accessed February 5, 2021).
10. NPR. Congress Overturns Internet Privacy Regulation. 2017. http://www.npr.org/2017/03/28/521831393/congress-overturns-internet-privacy-regulation (Accessed February 5, 2021).
11. Ren J, Lindorfer M, Dubois DJ, Rao A, Choffnes D, Vallina-Rodriguez N. Bug fixes, improvements, ... and privacy leaks: a longitudinal study of PII leaks across android app versions. San Diego, CA, USA: Network and Distributed Systems Security (NDSS) Symposium. 2018. ISBN 1-1891562-49-5. DOI: 10.14722/ndss.2018.23143
12. Mercurio M, Larsen M, Wisniewski H, Henson P, Lagan S, Torous J. Longitudinal trends in the quality, effectiveness and attributes of highly rated smartphone health apps. *Evid-Based Mental Health* 2020; 23: 107–11.
13. Blenner SR, Köllmer M, Rouse AJ, Daneshvar N, Williams C, Andrews LB. Privacy policies of android diabetes apps and sharing of health information. *JAMA* 2016; 315 (10): 1051–2.
14. Grundy Q, Chiu K, Held F, Continella A, Bero L, Holz R. Data sharing practices of medicines related apps and the mobile ecosystem: traffic, content, and network analysis. *BMJ* 2019; 364: l920.
15. He D, Naveed M, Gunter CA, Nahrstedt K. Security concerns in android mHealth apps. *Washington, DC, USA: American Medical Informatics Association Annual (AMIA) Symposium* 2014, pp. 645–654.
16. Huckvale K, Prieto JT, Tilney M, Benghozi P-J, Car J. Unaddressed privacy risks in accredited health and wellness apps: a cross-sectional systematic assessment. *BMC Med* 2015; 13 (1): 214.
17. DrWeb. Airpush exploited to spread trojans onto android devices. 2013. https://news.drweb.com/show/?i=3378&lng=en,202013
18. Ikram M, Ali Kaafar MA. 2017. A first look at mobile ad-blocking apps. In: proceedings of the Network Computing and Applications (NCA), Cambridge, MA, USA: 2017 IEEE 16th International Symposium*;* 30 Oct.-1 Nov. 2017. DOI: 10.1109/NCA.2017.8171376.
19. Ikram M, Vallina-Rodriguez N, Seneviratne S, Kaafar MA, Paxson V. An analysis of the privacy and security risks of android vpn permission-enabled apps. In: *Proceedings of the 2016 Internet Measurement Conference (IMC '16)*. New York, NY, USA: Association for Computing Machinery; May 13–17, 2019. 2016. pp. 349–364. DOI: 10.1145/2987443.2987471
20. Tang Z, Tang K, Xue M, *et al.* iOS, Your OS, Everybody's OS: Vetting and Analyzing Network Services of iOS Apps. In: proceedings of the 29th Usenix Security Symposium (Usenix Security). 2020.
21. Apktool. A tool for reverse engineering Android apk files. 2020. https://ibotpeaches.github.io/Apktool/ (Accessed February 5, 2021).
22. MD5Online. 3 reasons why md5 is not secure. 2020. https://www.md5online.org/blog/why-md5-is-not-safe/ (Accessed February 5, 2021).
23. Oracle. keytool. 2019. https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html (Accessed September 18, 2019).
24. VirusTotal Inc. Virustotal public api. 2019. https://www.virustotal.com/en/documentation/public-api/ (Accessed February 5, 2021).
25. Ikram M, Masood R, Tyson G, Kaafar MA, Loizon N, Ensafi R. In: Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA: ACM, New York, NY, USA. 10.1145/3308558.3313521
26. Kantchelian A, Tschantz MC, Afroz S, *et al.* Better malware ground truth: Techniques for weighting anti-virus vendor labels. In: Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security (AISec '15). New York, NY, USA: Association for Computing Machinery; 2015. pp. 45–56. DOI: 10.1145/2808769.2808780
27. Kharraz A, Robertson W, Balzarotti D, Bilge L, Kirda E. Cutting the Gordian knot: A look under the hood of ransomware attacks. In: Almgren M., Gulisano V., Maggi F. (eds) Detection of Intrusions and Malware, and Vulnerability Assessment. vol. 9148, Springer, Cham: Lecture Notes in Computer Science; 2015. DOI: 10.1007/978-3-319-20550-2_1
28. Arp D, Spreitzenbarth M, Hübner M, Gascon H, Rieck K. Drebin: effective and explainable detection of android malware in your pocket. In: proceedings of the Conference: Network and Distributed System Security Symposium (NDSS). San Diego, California, USA: 21st Annual Network and Distributed System Security Symposium (NDSS), February 23-26, 2014, 2014.
29. Bartel A, Klein J, Traon YL, Monperrus M. Automatically securing permission-based software by reducing the attack surface: an application to android In: 2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering. New York, NY, USA: Association for Computing Machinery, 2012. pp. 274–277. DOI: 10.1145/2351676.2351722
30. Porter Felt A, Chin E, Hanna S, Song D, Wagner D. Android permissions demystified. In: *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS '11*. New York, NY: Association for Computing Machinery; 2011: 627–638.
31. Au YKW, Zhou YF, Huang Z, Lie D. Pscout: Analyzing the android permission specification. In: *Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS '12*. New York, NY: Association for Computing Machinery; 2012: 217–228.
32. Android Permissions. 2019. http://developer.android.com/guide/topics/security/permissions.html (Accessed February 5, 2021).
33. mitmproxy - an interactive HTTPS proxy. 2019. https://mitmproxy.org (Accessed February 5, 2021).
34. Understanding Dual Stacking of IPv4 and IPv6 Unicast Addresses. 2020. https://www.juniper.net/documentation/us/en/software/junos/is-is/topics/concept/ipv6-dual-stack-understanding.html (Accessed February 5, 2021).
35. Ren J, Rao A, Lindorfer M, Legout A, Choffnes D. *ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic*. New York, NY: Association for Computing Machinery; 2016.
36. Zheng H, Li D, Liang B, *et al. Automated Test Input Generation for Android: Towards Getting There in an Industrial Case*. In: Proceedings of the 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP). Buenos Aires, Argentina: IEEE Press; pp. 253–262. 2017. DOI: 10.1109/ICSE-SEIP.2017.32
37. Aapa k Totkay. 2020. https://play.google.com/store/apps/details?id=com.naushad.aapaktotokay (Accessed February 5, 2021).
38. Health Benefit. 2020. https://play.google.com/store/apps/details?id=com.useofcocounutoil107 (Accessed February 5, 2021).
39. Nursing Care. 2020. https://play.google.com/store/apps/details?id=com.afra.nursingcareplans (Accessed February 5, 2021).
40. Infection Prevention. 2020https://play.google.com/store/apps/details?id=com.sigmatech.yadav.medical (Accessed February 5, 2021).
41. Sebastián M, Rivera R, Kotzias P, Caballero J. Avclass: A tool for massive malware labeling In: *International symposium on research in attacks, intrusions, and defenses (RAID). Springer, Cham: Telecom SudParis, France; pp. 230-253. 2016*
42. Hurier M, Suarez-Tangil G, Kumar Dash S, *et al.* Euphony: harmonious unification of cacophonous anti-virus vendor labels for android malware. In: *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*: IEEE Press; 2017: pp. 425–35. DOI: 10.1109/MSR.2017.57
43. Kaspersky. What is adware: What you should know and how to protect yourself. 2020. https://www.kaspersky.com.au/resource-center/threats/adware (Accessed February 5, 2021).
44. Kaspersky. What is Trojan virus. 2020. https://www.kaspersky.com.au/resource-center/threats/trojans (Accessed February 5, 2021).
45. Gunpoder - NJCCIC threat profile - official site of the state of New Jersey. 2016. https://www.cyber.nj.gov/threat-center/threat-profiles/android-malware-variants/gunpoder (Accessed February 5, 2021).
46. Third-party android app stores aren't all bad news. 2020. https://www.infosecurity-magazine.com/opinions/thirdparty-android-app-stores/ (Accessed February 5, 2021).
47. Zhang H, Yao D, Ramakrishnan N. Causality-based sensemaking of network traffic for android application security. In: Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security. New York, NY,

USA: Association for Computing Machinery, 2016. pp. 47–58. DOI: 10.1145/2996758.2996760

48. Malwarebyte Labs. 2020. https://blog.malwarebytes.com/detections/android-fakeapp/ (Accessed February 5, 2021).
49. Fakeapp - NJCCIC threat profile - official site of the state of New Jersey. 2018. https://www.cyber.nj.gov/threat-center/threat-profiles/android-malware-variants/fakeapp (Accessed February 5, 2021).
50. Trojan.gen.2 description - enigma soft. 2020. https://www.enigmasoftware.com/trojangen2-removal/ (Accessed February 5, 2021).
51. Wang H, Liu Z, Liang J, *et al*. Beyond Google Play: A large-scale comparative study of Chinese android app markets. In: *Proceedings of the Internet Measurement Conference 2018, IMC '18*. New York, NY: Association for Computing Machinery; 2018: 293–307.
52. Huawei Health. 2020. https://play.google.com/store/apps/details?id=com.huawei.health (Accessed February 5, 2021).
53. MyWellness. 2020. https://play.google.com/store/apps/details?id=com.technogym.mywellness (Accessed February 5, 2021).
54. Ren J, Rao A, Lindorfer M, Legout A, Choffnes D. Demo: ReCon: Revealing and controlling PII leaks in mobile network traffic. In: MobiSys 2016 - Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services Companion (MobiSys '16 Companion). New York, NY, USA: Association for Computing Machiner; 2016.p. 117. DOI: 10.1145/2938559.2938563
55. Papageorgiou A, Strigkos M, Politou E., Alepis E, Solanas A, Patsakis C. Security and privacy analysis of mobile health applications: the alarming state of practice. *IEEE Access* 2018; 6: 9390–403.
56. Dehling T, Gao F, Schneider S, Sunyaev A. Exploring the far side of mobile health: information security and privacy of mobile health apps on iOS and android. *JMIR mHealth and uHealth* 2015; 3 (1): e8.
57. Jon O, Miller C. Dissecting the android bouncer. *SummerCon2012, New York* 2012; 95: 110.
58. eweekdev. Google Bouncer Vulnerabilities Probed by Security Researchers. 2012. https://www.eweek.com/security/google-bouncer-vulnerabilities-probed-by-security-researchers/ (Accessed February 5, 2021).
59. Whitwam R. Circumventing Google's Bouncer, Android's anti-malware system. 2012. https://www.extremetech.com/computing/130424-circumventing-googles-bouncer-androids-anti-malware-system (Accessed February 5, 2021).
60. Maier D, Müller T, Protsenko M. Divide-and-conquer: Why android malware cannot be stopped. In: *2014 Ninth International Conference on Availability, Reliability and Security (ARES)*. Fribourg, Switzerland: IEEE; 2014: pp. 30–39. DOI: 10.1109/ARES.2014.12
61. Ren J, Lindorfer M, Dubois DJ, Rao A, Choffnes DR, Vallina-Rodriguez N. Bug fixes, improvements, ... and privacy leaks: A longitudinal study of pii leaks across android app versions. In: Proceedings of Network and Distributed System Security Symposium. 2018.
62. Zubaydi F, Saleh A, Aloul F, Sagahyroon A. Security of mobile health (mHealth) systems. In: Proceedings of the 2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE). Belgrade, Serbia: IEEE Computer Society; November 2-4, 2015, 2015. DOI: 10.1109/BIBE.2015.7367689
63. Urdu Best Totkays. 2020. https://play.google.com/store/apps/details?id=com.abc.urdubesttotkaabc (Accessed February 5, 2021).
64. FarmAlicante. 2020. https://play.google.com/store/apps/details?id=es.likisoft.farmalicante (Accessed February 5, 2021).
65. Yoga for Diabetes. 2020. https://play.google.com/store/apps/details?id=appinventor.ai_homestudioapp. yogafordiabities (Accessed February 5, 2021).
66. TritionRx. 2020. https://play.google.com/store/apps/details?id=com.tritionrxpro (Accessed February 5, 2021).
67. GuardianAngel. 2020.https://play.google.com/store/apps/details?id=appinventor.ai_homestudioapp. CommwithGuardianangel (Accessed February 5, 2021).
68. Smoke'n Vap'z. 2020. https://play.google.com/store/apps/details?id=com.appsvision.smokenvapz (Accessed February 5, 2021).
69. Fit Bites. 2020. https://play.google.com/store/apps/details?id=com.fit.bites (Accessed February 5, 2021).
70. COMM. 2020. https://play.google.com/store/apps/details?id=com.appsvision.comm (Accessed February 5, 2021).
71. iCom. 2020. https://play.google.com/store/apps/details?id=com.smartteam.icom (Accessed February 5, 2021).
72. OptiKoncept. 2020. https://play.google.com/store/apps/details?id=com.appsvision.optikoncept (Accessed February 5, 2021).
73. Cellu hit. 2020. https://play.google.com/store/apps/details?id=com.appsvision.celluhit (Accessed February 5, 2021).
74. Your Angels. 2020. https://play.google.com/store/apps/details?id=appinventor.ai_homestudioapp.angelsexist (Accessed February 5, 2021).
75. Vap'Pause. 2020https://play.google.com/store/apps/details?id=com.appsvision.vappause (Accessed February 5, 2021).
76. Doctor Street. 2020. https://play.google.com/store/apps/details?id=com.kenmac.DoctorMap (Accessed February 5, 2021).
77. Esthetic Medicare. 2020. https://play.google.com/store/apps/details?id=com.appsvision.estheticmedicarecenter (Accessed February 5, 2021).
78. Fragerstrom's TTest. 2020. https://play.google.com/store/apps/details?id=com.JJRRSoft.fragerstorm (Accessed February 5, 2021).