

THE IMPACT OF ANDROID MALWARE ON MOBILE HEALTH APPLICATIONS (mHealth Apps) SERVICES

Benjamin Aruwa Gyunka

Directorate of Information and Communication Technology
National Open University of Nigeria (NOUN)
Abuja, Nigeria
gyunkson@gmail.com

Abikoye Oluwakemi Christiana

Department of Computer Science
University of Ilorin
Ilorin, Nigeria
Kemi_adeoye@yahoo.com

Abstract— Sophisticated and advanced malware are being created on a daily basis and have become one most stealthy and dangerous attack approach used against critical information technology infrastructures in recent times. This paper seeks to expose Android malware attack strategies and concealment techniques and the consequent impacts on the security and privacy of electronic healthcare (eHealth) systems, particularly mobile health applications (mHealth apps). The main aim is to reveal the danger ignorant and vulnerable users, especially the sick and desperate, are exposed to when it comes to issues of Trust and Privacy arising from the use of these prevalent technology. A detailed analysis, based on secondary literatures, of various malware propagation methods and infection vectors was conducted. The research revealed that the healthcare industry information technology infrastructures experience the most of cyber attacks in recent years above other industries. With the adoption of ICT in Medicare, hospitals now accrues and stores more sensitive and personal data about their patients, more than accrued by banks and employers, but gives little or no attention to the security of their information systems. Malware attacks poses serious threats to information management and can distort critical health services such as the control of communicable and non-communicable diseases. The paper concludes by recommending for extensive peer-review of mHealth apps and for intensive measures to be taken by hospitals in implementing and continuous monitoring of a secured ICT infrastructure.

Keywords— *Android, Malware, mHealth, eHealth, Information Security, Communicable Disease, Non-Communicable Disease, information infrastructure, Operating System (OS)*

I. INTRODUCTION

The advent and rapid growth and adoption of information and communication technology (ICT) have swept over almost every sectors of human endeavour. Android is the most popular information technology platform in circulation today among both mobile devices and desktop computers [1]. Mobile devices, particularly the Smartphones and Tablets, have grown to become the closest companion and the first point of contact to their users in almost every situation. The Android operating system is more prevalent with a proliferation rate of 86.2% above all others mobile platforms [2] and [3]. Table 1 shows the different proliferation rates of various mobile platforms.

Table 1: Mobile platform proliferation rate [3]

Operating System	2Q16 Units	2Q16 Market Share (%)	2Q15 Units	2Q15 Market Share (%)
Android	296,912.8	86.2	271,647.0	82.2
iOS	44,395.0	12.9	48,085.5	14.6
Windows	1,971.0	0.6	8,198.2	2.5
Blackberry	400.4	0.1	1,153.2	0.3
Others	680.6	0.2	1,229.0	0.4
Total	344,359.7	100.0	330,312.9	100.0

However, the emergence of information technology, and the popularity of the Android platform has also come along with targeted attacks against both the information stored in these technologies and the technological systems themselves. Confidentiality, Integrity, Availability, and

Non-Repudiation is the core Model designed for protecting and securing information systems. This model has however been under persistent and continuous attacks by various cyber criminals who are interested in either stealing or sabotaging vital information or information assets for different reasons. Sophisticated and advanced malware are being created on a daily basis and have become one most stealthy and dangerous attack approach used against critical information technology infrastructures in recent times.

Information and Communication Technology is used as a great tool that increases productivity, reduces cost, and saves time across Institutions, especially the health sector [2]. Android devices are very strong and viable tool for accessing and communicating sensitive health related issues for their users [3]. Android mHealth apps are intended to serve as on-demand assistance. [2] outlined a list of top ten Android mHealth applications to include:

- CareZone: - an all-in-one tool that enables users oversee their family's medication and additional required doctor's instructions
- Doctor on Demand: - tool that enables users connect and board certified doctors in their area
- Doximity: - platform for inter doctor relations that enable them share and communicate about patient care and relevant medical news
- GoodRx Drug Prices and Coupons: - enable users compare prices and find better deals for their medications
- mySugr Diabetes Logbook: - tool that enable diabetics keep track of their health conditions
- Read by QxMD: - app that deliver medical related news, the latest medical research, topics reviews, and other medical-related literature
- Epocrates: - It contains lots of info about prescription and over-the-counter drugs, including proper dosage, the effects when combined with other drugs, possible side-effects and contraindications (factors that increases risks)
- WebMD: - it is a consult on the go app for medical questions that need answers but can't get to the doctor
- Medscape: - designed to provide vital medical information

Considering the mission of the healthcare industry to save lives, one would think that they will be free from cyber attacks. It is very easy to conclude that cyber criminals will not be interested on attacking health digital infrastructures and the vulnerable and desperate patients needing medical assistance. However, it is in the nature of criminals to take advantage of any kind of existing vulnerability and weakness to carry out their exploit against any individual or organizations. Cyber criminals are more subtle and swift at exploiting any given vulnerability found on an internet network or any information systems. Malware creation is

one of the most subtle and technical attack approaches adopted against the Android platform. Given that mobile devices are mostly personalised, very wealthy and influential patients, doctors and other professional health providers are the prime targets of this malware attacks. This paper seeks to take a deeper look into Android malware and their effects on providing efficient digital healthcare services especially in the area of managing information and controlling communicable and non-communicable diseases.

II. RELATED WORK

Smartphones and Tablets are cutting-age technological advancement of the 21st Century which has unequivocally introduced rising security challenges in the field of computing. The emergence and proliferation of these Hand-held Devices is almost reaching an epidemic proportion [6]. The world's population study reveals the existence of about seven billion people in the Universal [7] and more than 87% of this population already have been penetrated by the mobile market [8]. These devices have the in-built capabilities of remaining constantly connected to the internet (due to the possession of features such as Wi-Fi, voice, data, GPS, etc.) and have continued, at an exponential rate, to win the confidence and trust of their users. Their emergence has simplified lots of social networking activities, commercial and banking transactions so much that they have become an integral part of our lives. [9, p. 37 & 38] observed that the trust that people have in their handheld devices has by far outweigh the trust they have for their fellow humans. These devices hold more of their intimate personal secrets and corporate life that no one else has access to. This has created wider vulnerable environments which have been rapidly exploited, on a daily basis, by different category of malicious users and hacktivist. Malicious activities and attacks have continued to grow exponentially against mobile devices especially those running the Google Android OS, which is an open-source software stack, given their popularities and influence in the economy today [10] and [11]. The threats confronting these devices can be Application Based, Network Based and Web Based [12]. Attacks through malicious applications are more prominent against the Android Platform [13]. Thus, mHealth applications and eHealth have serious privacy and security issues [14], [15], [16], [5], [17], [18], and [19]. It has become very imperative therefore to critically research Android malware, their behaviours and concealment strategies which they deploy against information systems.

III. THE ANDROID FRAMEWORK

The android platform is divided into three main parts namely, the Software Development Kit (SDK); the Operating System (OS); and Applications [20]. This platform is built upon Linux 2.6 kernel and was originally under the supervision of the Open Handset Alliance [21].

Because of this open source nature of the platform, lots kinds of developers were attracted to it [22]. The Linux kernel is the bedrock of the Android platform. Every essential components and services, such as process management, security, memory management, and lots more are only enabled and controlled by the Linux kernel. The Android framework comprises of different layers which begin with the Linux Kernel at the very bottom, forming the core of the operating system. Another key component in this framework is the Dalvik Virtual Machine. The Dalvik VM creates an enabling environment that allows processes to run separately [23] and [24], each having its unique process id either known as User ID (UID) or Group ID (GID) and also having unique storage [25]. The existence of UID and GID disallow application processes from accessing or sharing the storage or resources of each other without strict permission granted [22].

Software Development Kits (SDK) is a Java toolbox which Google have released freely to the public to enable anyone creates any Android application at will without restrictions [22]. One of the most important tool found in the SDK is the Android Debug Bridge (ADB), it helps to interface an Android device with a computer system to enhance effective communication and application development [27]. Through the adb the application developer is able to access the command shell via the computer system and other activities such as application installation/uninstallation, reading of logged files, and file transfers and be easily made without obstructions [20].

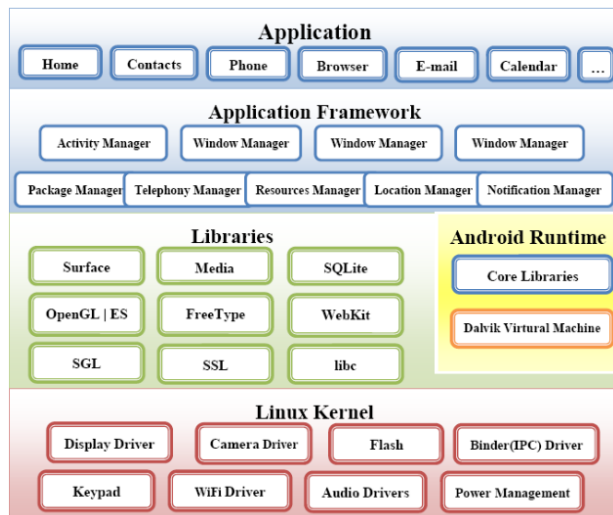


Figure 1: Android Framework [26]

The operating system (OS) makes use of the concept of sandbox to manage applications and processes execution. It uses also the SQLite database including several file system formats like YAFFS2 (Yet Another Flash File System 2), RFS, and EXT4 [20] and [25]. Android Operating System has now extended beyond just running on Smartphones and

Tablet but it is found on net books, e-readers, cars, and home and hospital appliances [22].

A. Android Data Storage Directories

Mobile phones are traditionally known to store important data in SIM-cards or device's internal memory [28] or even with the mobile network providers [29] and in an external memory card. Android's internal memory is called NAND memory. The Android stores data in directories as shown below [30];

- Share Preferences: - the data here is stored as key-value pairs which found in the directory "/applications/data/shared_pref"
- Internal Storage: - this storage is also known as the NAND flash, it contains mostly private sensitive data
- External Storage: - the directory for this storage is '/sdcard', it contains all public information and does not necessarily have strong security measures
- SQLite: - SQLite is a structured Android database with the directory '/data/data/Package/database

The concept of rooting an Android device is a delicate and very risky process. Rooted devices are highly vulnerable and exposed; malware can easily penetrate the filesystem, defiling all security measures, to access all these data. More so, the adb tool can be used against the rooted device by an attacker to extract almost all allocated data on the system [31].

B. Android Application Framework

All Android applications are created and compiled as Android Package (APK) file [32] and [33]. The APK file is simply a ZIP format archive file that is renamed to have an .apk extension [34]. APK files would normally have to be unzipped, as the case with normal zip files, in order for its contents to be extracted. The content consists of a Dalvik executables, resources, native libraries and a manifest file, and is usually signed by the developer of the application using self-signed certificate [33] and [32]. In particular, the APK file would usually contain two folders (META-INF and res) and three files (classes.dex, AndroidManifest.xml, resources.arsc) as can be seen in figure 2 [32]. The classes.dex and AndroidManifest.xml are the most delicate and important components of the APK file which are usually the high targets of malware creators [35] and [36]. The classes.dex is the Dalvik Virtual Machine executable file which contains the main working code of the application. That is, the payloads of an app is created and defined in classes.dex file. The AndroidManifest.xml provides semantic-rich information about the application which includes the version and required permissions governing an app's operations. AndroidManifest.xml also contains four basic components which are Activities, Services and Broadcast Receivers and Content Providers. It

is important to note that without the information in manifest file, it is impossible for an application to run on the Android OS [33] and [35]. Attackers target these two files to either inject malicious code [36] or alter the permissions for nefarious purposes. Activity is simply an element of the user interface which represents a screen. Service works at the background and can run indefinitely.

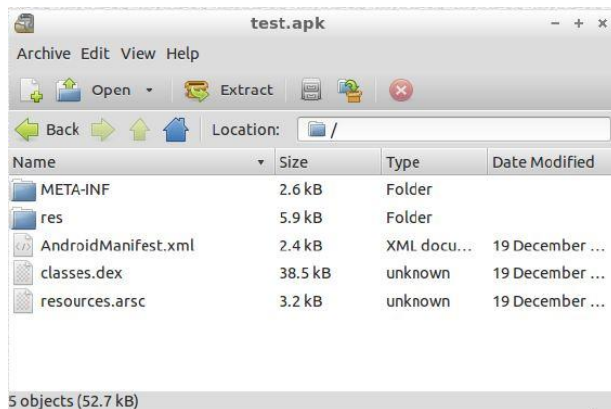


Figure 2: An APK unzipped file displaying its contents [34]

Broadcast receiver is an application component which receives broadcast messages and starts a workflow according to the obtained message. Content Provider is the component that provides an application with the abilities to store and retrieve data. It also enables sharing a set of data with another application. Importantly, the different components contained in an Android Package makes an app not to have a central entry point like in the case of Java programs that have main method. Not having a central entry point implies that all components (except for broadcast receivers which may be dynamically defined also) must necessarily be declared by the application's author in the AndroidManifest.xml file [33].

IV. ANDROID MALWARE

Malware is any software used to disrupt computer operation, perform unauthorized functions, gather sensitive information or gain access to private information systems [37]. Malware can range from a simple nuisance such as pop-up advertising to an intrusion causing severe damage such as stealing passwords and data or infecting other machines on a network. Malware appear in different forms such as Viruses, Spyware, Adware, Nagware, Backdoors, Rootkits, Trojans Horse, Botnets keyloggers, spammer and Flooder, Zombie, Auto-rooter, Logic bomb, and Worms [38] and [39]. Within the mobile platform, malware are able to propagate between devices using infection vectors such as Bluetooth, MMS, SMS, Internet networks, executable files, emails and web vulnerabilities [1]. Other infections vectors include social engineering, synchronization/USB, WiFi, Device Rooting, Near Field Communication (NFC) Peripheral memory cards, Drive-by downloads, and Google Play Store [38].

Malware can be classified according to their behaviour and payloads [32]. They can also be classified in accordance to the way they spread or propagate [38]. There is the group that need to be triggered by a program or by some users' activities to enable them replicate and spread (Such as viruses), the second group of malware (Worms) are self replication ability and can automatically scan for vulnerable victims through internet networks or any medium in contact with them. The third category evolves into exploits and unknown vulnerabilities. They employ multiple attack vectors in order to replicate. These last categories are known to strike faster, preventing timely intervention by security administrators. Some notable Android propagation technique include Drive-by Download, Repackaging [40]. Devices that are infected with Malware, usually, are largely under the control of the attackers. They can provide an unauthorized gateway for the attacker to be able to access and steal very personal sensitive information such as Bank details, confidential messages, files, photos and perform online/click fraud and phishing. The attacker will also be able to remotely control the device by placing illegal phone calls, sending spam emails, SMS, and MMS. "Fake Player" was the first Android Malware discovered in 2010 then subsequently others such as Pjapss, Hippo, Geinimi, DroidDream, Droidkungfu and Basebridge and NotCompatible, which is an Android Web Proxy Bot. KorBanker is another example of a nasty Android malware found in South Korea which targeted several popular South Korean banking apps. The most recent Android malware released in the wild is Stagefright [41]. Android malware are always on the increase and are becoming tougher and resistance, through different concealment techniques and sophistications, to existing detection methods and models [38]. Android malware behave differently in accordance to their stated payloads and requires different requisite techniques to put them in check [42].

The kind of attacks deployed by malware includes unauthorized access, masquerading, denial of service, annoyance attack, and eavesdropping [38]. Malware have evolved with advanced attack techniques against the Confidentiality, Integrity, and Availability Model of information security. The confidentiality, integrity, and availability of information and information assets can no longer be guaranteed with the sophistication evolvment in malware developments and attacks. The main idea behind malware development is to generate revenue for the developer (attacker) while generating cost for the user (victim). This is achieved as the malware are targeted at stealing financial information (Credit/Debit Card details, Mobile banking Login credentials, and Hijacking e-commerce transaction sessions) and other very personal information via click fraud, extortion and espionage. Usually, malware are mostly operating systems and hardware specific, but recent development has seen the creation of advanced malware that have the capacity to

operate across platforms –especially between Android and Windows which are the most popular mobile and desktop platforms [38]. This simply implies that there exist malware that can now successfully operate across two or more distinct operating systems.

A. Malware Concealment Strategies

Malware advanced sophistications are aimed at enabling escape and withstand any existing detection methodologies. The ability of a malware to hide its presence and escape detection is also termed as concealment strategies. The goal of this strategy is to enable the mal-code survive longer without detection in order to have ample time available to replicate and infect more devices. That is, the technique increases the life span of a malware between infection and detection phases. Also, these techniques aim to make the analysis part of malware more difficult for the antimalware developers. [38] noted that malware concealment techniques can be classified into two main groups. The first group bypasses anti-virus or anti-malware signature detection by implementing a set of methods to enable it hide itself. Group two are not only able to hide their presence from detection but also can make active effort to hinder the detection and analysis of the code using anti-emulation, anti-debugging techniques and anti-disassembling techniques and also evade signature-based detection methods. These second groups of malware are known to implement aggressive coding techniques for self-protection while the first group does not have the ability to attempt to defend itself. The first class of malware adopts a passive concealment approach while the second class deploys an active approach. Thus, malware passive concealment techniques involve code obfuscation, entry point obfuscation, encryption, Packing, Oligomorphism, Polymorphism, and Metamorphism. Active strategies for malware concealment involve Anti-emulation, Anti-debugging, and Anti-disassembling.

1) Active Approaches

a) *Anti-emulation*: Across all platforms, the use of virtual machines and emulation are very vital aspect of malware analysis because they provide a safe and accurate environment for the analysis and evaluation of real life behaviour of malware and its impact on the targeted system. The use of emulation as safe haven for malware analysis by security analyst has also received counter response from malware creators who have in turn implemented anti-emulation techniques to counter analysis and evade detection.

b) *Anti-debugging*: An anti-debugging technique is mostly deployed to trick debuggers, via code obfuscation technique, which then makes the job of anti-malware analysis much more difficult.

c) *Anti-disassembling*: Anti-disassembly technique employs special code within a file that trick the disassembly

and analysis tools into producing incorrect programme listings. This technique provides an additional layer to the extensive array of armour available for malware creators. When implementing anti-disassembling in their code, malware authors have two common goals they strive to achieve – that the function should not be easily automated and that the source code should not be available until the code runs.

2) Passive Approaches

a) *Obfuscation*: Obfuscation is a method adopted by code developers to enforce a security policy targeting at protecting the code from being analysed and understood. Code obfuscation is a deliberate modification of the code with the intention of increasing its detection difficulty by anti-malware and understanding by humans. Code obfuscation can successfully be done through dead code insertion, code transposition and registers reassignment (this technique is extensively used by metamorphic viruses). Entry point obfuscation is another technique which viruses hidden inside executable files adopt to call their malicious code. When a file is infected by a virus, the virus seeks for a way to take control of the file and inject its malicious code. the payload of such malware are usually hidden inside unusual places of the executable file and thus tries to distract the anti-virus or anti-malware engine from investigating the file. Entry point obfuscation is considered as an anti-heuristic method and is often deemed hard to detect, disinfect and remove.

b) *Encryption*: This is a technique adopted by most malware that involves scrambling their contents or code. The malware will usually create a random encryption key which it then stores in the body of the malware and then encrypts its remaining part. It comes alive when invoked and it then decrypt itself for action [39].

c) *Packing*: author and affiliation lines of affiliation 1 and copy this selection.

d) *Oligomorphism*: Insert one hard return immediately after the last character of the last affiliation line. Then paste down the copy of affiliation 1. Repeat as necessary for each additional affiliation.

e) *Polymorphism*: Place your cursor to the right of the last character of the last affiliation line of an even numbered affiliation (e.g., if there are five affiliations, place your cursor at end of fourth affiliation). Drag the cursor up to highlight all of the above author and affiliation lines. Go to Column icon and select “2 Columns”. If you have an odd number of affiliations, the final affiliation will be centered on the page; all previous will be in two columns.

f) *Metamorphism*: Metamorphic malware employs the technique of swapping registers in a system or CPU from one form or generation to another in order to prevent conventional and known signature and pattern matching from working. This malware have the ability to re-programme itself as it evolved after each successful

infection. It does this by completely rewriting itself at each iteration which increases the difficulty of detection [39]. Metamorphic malware is divided into two main groups. The first group stages out the communication capability of the malware whether it is capable of open-world communication (downloading further plugins and updating itself with new features) or closed-world communication with no communication capability. The second group is characterized by the transformation capability of the malware into different behaviour and appearance. Metamorphic malware is believed to be more complex than any other malware [38], for the possesses a strong ability to dynamically disassemble themselves, transform their code, and reassemble themselves into an executable form.

B. Trust and Privacy in e-HealthCare

Health problems drive people to seeking for varied information from and through various sources in order to deal with their health challenges [14]. With the advent of the internet (Information Technology) and particularly mobile devices, the amount of information within the finger tip of health patients and professionals is of great volume so much that they are faced with the challenge of Trust and Security for these technologies [14] and [15]. The wide spread and adoption of Smartphones and the corresponding large volume of various mobile health application in circulation [44], more and more people now share their sensitive and personal healthcare information using these apps [45]. Users' private information is left at the mercies of the developers of these apps as the common users usually do not know how these applications manage or use their data. This poses serious security, privacy and trust issues to the general public. This is because there is a lack of substantive reference information for consumers to base well-informed decisions about whether or not to adopt the applications they review and to ascertain the validity of the information provided by these e-health solutions [14]. With the exponential rise of malicious mobile apps, it is becoming increasingly difficult to guarantee the quality of these m-health apps, particularly in respect to serious concerns regarding the safety and validity of the information the common users' access through them [46] and [47]. [15] observed that up to date, there is just little scientific proof that mobile health applications (m-Health apps) can be trusted and used safely.

A most recent report shows that among all America's critical infrastructures, the healthcare industry is the most targeted and plagued by perpetual persistent attacks from numerous unknown malicious hackers whose intents is to exploit vulnerabilities in their insecure and antiquated networks in order to exfiltrate patient health records [17]. The record also reported that approximately 47% of United State populations have experienced a compromised in their personal healthcare data over the last 12 months. This goes on to show that the healthcare sector suffered from the most recent data breaches. It is amazing the broad amount of

sensitive patients' data collected by hospitals, this includes address, social security numbers, spouse, children, credit cards and bank accounts [17]. Research has shown also that hospitals accrues a surprisingly wide amount of information and stores it in one (often-vulnerable) system [17]. The domain of medical mobile application development has thus become a high target for malware developers who are either after the susceptible users for personal gains or for espionage. It can clearly be stated that your hospital has a greater and broader amount of your private data than your employer or your bank does [17]. The Healthcare industry is extremely susceptible to cyber-attacks and very little fraction of them are given attention to implementing strong information security systems on their infrastructures [17].

The health sector is so engulf and very busy, channeling all their resources, to fulfilling their main vision, which is saving lives, that they give little or no attention to the security of their Information Technology Systems. Malicious attackers on the other hand have seen this selfless dedication and commitment to saving lives as a weakness. Being fully aware of this strong weakness and vulnerability in the health industry, they are making due diligent exploiting it through various attack surfaces such as advanced persistent threats (APTs), Distributed Denial of Service (DDoS) attacks, malware infections, cyber espionage, and data and intellectual property theft [16]. Thus, the Integrity, Confidentiality, and Availability of m-health users' data cannot be completely assured [18]. It therefore recommended that the health sector would need to invest in risk management based information security programs and should engage in cyber security programs that deploys a multilayered defense that protects the confidentiality, integrity and availability of information whenever it is stored, in transit, or being processed [17].

C. The Challenge of Android Security Model

The Android operating system has in-built authentication mechanisms which enables the use of passwords, tactile patterns or biometric information [20]. Most importantly, the platform takes advantage of its file system security and permission model derived from its core Linux kernel, to enforce its own internal security [33]. As can be observed from figure 1, the Linux Kernel is the lowest level of the Android framework and it is the backbone of the Android inbuilt security. Every layer of the framework has some level of security they provide. Any further security measures that is designed for this platform can only be built upon existing ones [48]. A few security challenges confront Android devices. The first is that they have limited and restricting security architecture, such as slow processing power, small storage capacity and short battery power. These limited resources make it impossible for security experts to run a complete anti-malware solution in them. The second challenge is that most existing malware analysis techniques were built to reside and run at high level

within the operating system stack [38] and [42]. The performance of these detection techniques can easily be influenced by a compromised OS in which they reside. That is, an infected device can create an unreliable environment for the analysis tools by influencing their activities [1]. Attackers are fully aware of this loophole and have successfully implemented various concealment strategies, both active and passive to evade detection. The Domain of Android malware is a virile one that requires more extensive research into novel ways of curbing the emergence of new and sophisticated malware and protecting critical information infrastructures.

V. CONCLUSION

The rapid proliferation and adoption of Smartphones, especially the Android, has seen massive development of different mobile applications created to ease lots of human activities by reducing workloads, improve healthcare quality, improve efficiency, reducing cost and saving time [18]. Considering the importance of Smartphones and mobile apps in the healthcare sectors, it can be observed that physicians and other health professionals now use them to access patient records, to view test results, and to prescribe medications. Health patients also use these mobile health apps to access and update their medical records, to monitor their health statistics, and to also view their prescription [45]. Although these technologies are very helpful and handy tools for critical information management and dissemination especially in contributing to the control of communicable and non-communicable diseases in the health sectors, strong and extensive peer-review of these apps is necessary. Stronger security and privacy measures have to also be put in place to ultimately fight against cyber attacks that will take advantage of the vulnerable health patients through malware attacks and mismanagement. Priority has to be placed on Android platform because of its open nature and popularity; most of the apps developed for the platform do not require approval from anyone before they are released into the market [49]. This defines why Android malware ranges highest above malware of other platforms. If the issues of Trust and Privacy of Information and sensitive data can be addressed in the health industry, the benefits of information and communication technology will be greatly felt in regard to communicating relevant, life-saving, health information to the public.

References

- [1] K. Dunham, S. Hartman, J. A. Morales, M. Quintans and T. Strazzere, *Android Malware Analysis*, New York: CRC Press, 2015.
- [2] International Data Corporation, "Smartphone OS Market Share, 2016 Q2," 17 August 2016. [Online]. Available: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. [Accessed 30 September 2016].
- [3] Gartner, "Gartner Says Five of Top 10 Worldwide Mobile Phone Vendors Increased Sales in Second Quarter of 2016," 18 August 2016. [Online]. Available: <http://www.gartner.com/newsroom/id/3415117>. [Accessed 30 September 2016].
- [4] Health, "Top medical apps for Android," 29 August 2016. [Online]. Available: <http://androidapps.com/top-medical-apps-for-android/>. [Accessed 30 September 2016].
- [5] D. He, N. Muhammad, C. A. Gunter and N. Klara, "Security concerns in Android mHealth apps," in *In AMIA Annu Symp Proc*, Washington DC, pp. 645-654, 2014.
- [6] B. Mellars, "Forensic examination of mobile phones," *Digital Investigation*, vol. 1, no. 4, pp. 266-272, 2004.
- [7] United Nations, Department of Economic and Social Affairs, Population Division, "World Population Prospects: The 2010 Revision, Highlights and Advance Tables," United Nations (UN), New York, pp. 17-48, 2011.
- [8] C. Knudsen, "Smartphones, tablets and the mobile revolution," 29 January 2013. [Online]. Available: <http://www.mobilemarketer.com/cms/opinion/columns/14667.html>. [Accessed 18 April 2013].
- [9] A. Hoog, *Android Forensics: Investigation, Analysis and Mobile Security for Google Android*, J. McCash, Ed., Amsterdam: Syngress, 2011.
- [10] B. Baskaran and R. Anca, "A Study of Android Malware Detection Techniques and Machine Learning," in *MAICS 2016*, pp. 15-23, 2016.
- [11] S. Chatterjee, P. Kasturi, R. Reek and N. Asoke, "A Comprehensive Study on Security issues in Android Mobile Phone—Scope and Challenges," *International Journal of Innovative Research in Advanced Engineering*, vol. 3, no. 3, pp. 62-72, 2016.
- [12] E. Charani, E. Castro-Sánchez, L. S. Moore and A. Holmes, "Do smartphone applications in healthcare require a governance and legal framework? It depends on the application!," *BMC Medicine*, vol. 12, no. 1, p. 1, 2014.
- [13] F. Rafael, S. Julian and K. Marcel, "On the Effectiveness of Malware Protection on Android - An evaluation of Android antivirus," *Fraunhofer AISEC*, p. 45, 2013.
- [14] W.-C. Su, "A preliminary survey of knowledge discovery on smartphone applications (apps): Principles, techniques and research directions for e-health," in *ICME International Conference on Complex Medical Engineering* June 26-29, Taipei, 2014.
- [15] R. J. Riezebos, "Peer-reviewing of mHealth applications," *University of Amsterdam, Amsterdam*, pp. 6-46, 2014.
- [16] B. Filkins, "Health Care Cyberthreat Report: Widespread compromises detected, compliance nightmare on horizon," *SANS Institute*, Bethesda, MD, pp. 2-22, February 2014.
- [17] ICIT, "ICIT Brief Hacking Healthcare IT in 2016: Lessons the HealthCare Industry can learn from the OPM Breach," *Institute for Critical Infrastructure Technology*, pp. 1-11, January 2016.
- [18] D. Kotz, "A threat taxonomy for mHealth privacy," in *In COMSNETS*, Hanover, pp. 1-6, 2011.
- [19] KPMG, "Health Care And Cyber Security: Increasing Threats Require Increased Capabilities," in *KPMG 2015 Cyber Healthcare Survey*, pp. 1-7, August 2015.
- [20] M. d. L. S. Andre, C. S. Fabio, P. d. M. Laerte and T. d. S. J. Rafael, "Acquisition of digital evidence in android," in *Proceedings of the 9th Australian Digital Forensics Conference*, pp. 1-10, 2011.
- [21] J. Lessard and G. C. Kessler, "Android Forensics: Simplifying Cell Phone Examinations," *Small Scale Digital Device Forensics Journal*, vol. 4, no. 1, pp. 1-12, 2010.
- [22] A. Hoog, "An introduction to Android forensics," 2010. [Online]. Available: <http://www.dfinews.com/articles/2010/04/introduction-android-forensics?page=0%2C0&pid=974#UgPsHZbsZ3I>. [Accessed 08 August 2013].
- [23] M. Yates, "Practical Investigations of Digital Forensics Tools for Mobile Devices," in *2010 Information Security Curriculum Development Conference*, pp. 156-162, 2010.
- [24] M. Chow, *Composer, Android Forensics Demo*. [Sound Recording]. INFOSEC World Conference and Expo. 2012.
- [25] C. Triplett, "Android Forensics Background, techniques and analysis tools," 2013. [Online]. Available: <https://viaforensics.com/mobile->

- security/htcia-android-forensics-training-presentation-march-22-2013.html. [Accessed 04 August 2013].
- [26] L. Xinfang, Y. Chunghuang, C. Shihjen and W. Jainshing, "Design and Implementation of Forensic System in Android Smart Phone," in In The 5th Joint Workshop on Information Security., pp. 1-11, 2009.
- [27] Developers, "Android Debug Bridge," 2013. [Online]. Available: <http://developer.android.com/tools/help/adb.html>. [Accessed 27 August 2013].
- [28] S. Y. Willassen, "Forensic analysis of mobile phone internal memory," in Advances in Digital Forensics, pp. 191-204, 2005.
- [29] D. Forte and A. Donno, "Mobile Network Investigations," in Handbook of digital forensics and investigation, C. Eoghan, Ed., Amsterdam, elsevier Inc., 2010, pp. 517-557.
- [30] T. Rohit, "Android Forensics," 2013. [Online]. Available: <http://resources.infosecinstitute.com/android-forensics/>. [Accessed 18 August 2013].
- [31] S. Kevin, "ADB Method," 2011. [Online]. Available: <https://viaforensics.com/android-forensics/adb-method.html>. [Accessed 18 August 2013].
- [32] R. Paolo, "Developing A Next-Generation Mobile Security Solution For Android," Reykjavik University, pp. 91, 2014.
- [33] Y. Zhauniarovich, Android Security (and Not) Internals, Trento: ASANI, 2014.
- [34] P. O. Fora, "Beginners Guide to Reverse Engineering," in RSA Conference 2014, San Francisco, pp. 1-22, 2014.
- [35] S. Rushil, "Analyzing and Dissecting Android Applications for Security defects and Vulnerabilities," Blueinfy, pp. 1-14, 2011.
- [36] F. Tchakounté and P. Dayang, "System calls analysis of malwares on android," International Journal of Science and Technology, vol. 2, no. 9, pp. 669-674, 2013.
- [37] M. Federico, "Malicious Android Apps - Overview, Status and Dilemmas," Andrototal, pp. 1-75, 2014.
- [38] A. Nicholas, "Cross-Platform Malware Contamination," RHUL, London, pp. 1-84, 2015.
- [39] R. Indrajit, "Computer Security," Colorado State University, 2013.
- [40] Y. Zhou and J. Xuxian, "Dissecting android malware - Characterization and evolution," in In 2012 IEEE Symposium on Security and Privacy, pp. 95-109, 2012.
- [41] N. Phil, "The 'Stagefright' exploit: What you need to know," 17 August 2015. [Online]. Available: <http://www.androidcentral.com/stagefright>. [Accessed 15 October 2016].
- [42] R. Rahul, R. Venkiteswaran, J. B. Anoop and K. D. Soumya, "Android Malware Attacks and Countermeasures: Current and Future Directions," in Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on IEEE, pp. 137-143, 2014.
- [43] A. Sharma and K. S. Sanjay, "Evolution and detection of polymorphic and metamorphic malwares: a survey," p. 1406.7061, 2014.
- [44] M. Alexander, U. Philipp, W. Harald and S. Stefan, "Analyzing Privacy Risks of mHealth Applications," in Studies in health technology and informatics 221, pp. 41-45, 2015.
- [45] P. Miloslava, A. Steven and C. Samir, "A Taxonomy of mHealth Apps – Security and Privacy Concerns," in System Sciences (HICSS), 2015 48th Hawaii International Conference on, pp. 1-10, 2015.
- [46] B. J. Visser and W. B. Arthur, "Need to peer-review medical applications for smart phones," Journal of Telemedicine and Telecare, vol. 18, no. 2, pp. 124-124, 2012.
- [47] T. L. Lewis, "A systematic self-certification model for mobile medical apps," Journal of medical Internet research, vol. 15, no. 4, p. e89, 2013.
- [48] N. Elenkov, Android Security Internals: An In-Dept Guide to Android's Security Architecture, San Francisco: No Starch Press, Inc., 2015.
- [49] R. Meier, Professional Android 4 application development, John Wiley & Sons, 2012.
- [50] J. Morales, X. Shouhuai and S. Ravi, "Analyzing malware detection efficiency with multiple anti-malware programs," in Proceedings of 2012 ASE CyberSecurity'12, pp. 56-66, 2012.
- [51] A. H. Sung, X. Jianyun, C. Patrick and M. Srinivas, "Static analyzer of vicious executables (save)," in . In 20th Annual Computer Security Applications Conference, 2004, pp. 326-334, 2004.
- [52] H. Shewale, P. Sameer, D. Vaibhav and S. Pragya, "Analysis of android vulnerabilities and modern exploitation techniques," ICTACT Journal on Communication Technology, vol. 5, no. 1, pp. 863-867, 2014.
- [53] H. Gonzalez, A. A. Kadir, S. Natalia, A. J. Alzahrani and A. A. Ghorbani, "Exploring reverse engineering symptoms in Android apps.," in In Proceedings of the Eighth European Workshop on System Security, pp. 7, 2015.
- [54] O. Jon and M. Charlie, "Dissecting the Android Bouncer," in SummerCon 2012, pp. 61, 2012.
- [55] A. Desnos and G. Geoffroy, "Android: From reversing to decompilation," in Proc. of Black Hat Abu Dhabi, pp. 77-101, 2011.
- [56] DREAM LAB Technologies, "Reversing Android Apps," DREAM LAB Technologies, pp. 61, 2011.
- [57] A. D. Vicente, "Android reverse engineering: understanding third-party applications," The OWASP Foundation, Romania, pp. 1-24, 2013.
- [58] N. Godfrey, Decompiling Android, Apress, 2012.
- [59] S. Vanja, "Sophos Mobile Security Threat Report," SophosLabs, pp. 1-10, 2014.
- [60] Y. Zhou, W. Zhi, Z. Wu and J. Xuxian, "Hey, You, Get Off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets," in In NDSS, pp. 50-52, 2012.
- [61] R. Adhikari, D. Richards and K. Scott, "Security and Privacy Issues Related to the Use of Mobile Health Apps," in Proceedings of the 25th Australasian Conference on Information Systems, 8th - 10th December, Auckland, pp. 1-11, 2014.
- [62] F. Zubaydi, S. Ayat, A. Fadi and S. Assim, "Security of Mobile Health (mHealth) Systems," in In Bioinformatics and Bioengineering (BIBE) 2015 IEEE 15th International Conference on, pp. 1-5, 2015.
- [63] C. Zhang, Z. Xihui and H.-N. Richard, "Assessment Metrics, Challenges, and Strategies for Mobile Health Apps," Issues in Information Systems, vol. 14, no. II, pp. 59-66, 2014.
- [64] L. Ohno-Machado, W. Shuang, W. Xiaofeng, I. Arya and J. Xiaoqian, "Privacy, Security, and Machine Learning for Mobile Health Applications". aaas, pp. 1-13, 2014
- [65] The mHealth Alliance, "Patient Privacy In A Mobile World - A Framework To Address Privacy Law Issues In Mobile Health," mHealth Alliance, pp. 116, June 2013.
- [66] Y. Y. Yip, "There's an app for that: the use of mobile medical applications in clinical practice," Ontario Medical review, p. 48, 2014.
- [67] "Cyber Security Management: A Review," Business Management Dynamics, vol. 5, no. 11, pp. 16-39, 2016.
- [68] E. Chin, P. F. Adrienne, S. Vyas and W. David, "Measuring user confidence in smartphone security and privacy," in In Proceedings of the Eighth Symposium on Usable Privacy and Security, pp. 1-15, 2012.
- [69] G. Zyba, "Mobile Malware Propagation and Defense," University of California, San Diego, pp. 115, 2013.
- [70] A. John, "mobile Apps and mobile Devices Safe to Use?," NHS Ayrshire and Arran, pp. 1-23, 2015.
- [71] P. CARL, "Best Android Apps For Doctors, Physicians, And Medical Practitioners," 25 May 2012. [Online]. Available: <http://www.androidauthority.com/best-android-apps-doctors-physicians-health-professionals-88851/>. [Accessed 30 September 2016].
- [72] M. Pomilia, "A study on obfuscation techniques for Android malware," Sapienza University of Rome, pp. 81, 2016.
- [73] K. W. Hamlen, "Stealthy software: Next-generation cyber-attacks and defenses," in In Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on, pp. 109-112, 2013.
- [74] D. Maiorca, A. Davide, C. Igino, A. Marco and G. Giorgio, "Stealth attacks: An extended insight into the obfuscation effects on Android malware," Computers & Security, vol. 51, pp. 16-31, 2015.
- [75] K. Allix, J. Quentin, T. F. Bissyandé, K. Jacques, S. Radu and L. T. Yves, "A Forensic Analysis of Android Malware--How is Malware Written and How it Could Be Detected?," in In Computer Software and Applications Conference (COMPSAC), 2014 IEEE 38th Annual, pp. 384-393, 2014.