SMARTPHONE SECURITY: CLASSIFICATION AND CATEGORIZATION OF SMARTPHONE MALWARE (PROOF OF CONCEPT AND FOUND IN THE WILD)

By

Humayun Ali

2010-NUST-MSCCS-26

Supervisor

Dr. Zahid Anwar

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters in Computer and Communication Security (MS CCS)

In
School of Electrical Engineering and Computer Science (SEECS),
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(July 2014)

Approval

It is certified that the contents and form of the thesis entitled "**Smartphone Security: Classification and Categorization of Smartphone Malware (Proof of Concept and Found In The Wild)**" submitted by **Humayun Ali** have been found satisfactory for the requirement of the degree.

Supervisor: **Dr. Zahid Anwar**

Signature: _____

Date: _____

Committee Member 1: **Dr. Adnan Khalid Kiani**

Signature: _____

Date: _____

Committee Member 2: **Dr. Hamid Mukhtar**

Signature: _____

Date: _____

Committee Member 3: **Dr. Ijaz A. Qureshi**

Signature: _____

Date: _____

To

My Mother, Father

&

Niece

Certificate of Originality

I hereby declare that the thesis titled "**Smartphone Security: Classification and Categorization of Smartphone Malware (Proof of Concept and Found In The Wild)**" is my own work and to the best of my knowledge. It contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST or any other educational institute, except where due acknowledgment, is made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic is acknowledged. I also verified the originality of contents through plagiarism software.

Author Name: **Humayun Ali**

Signature: _____

Acknowledgment

I planned, He planned and He is the best planner. He didn't only plan but He also chose the best path that I could only dream or imagine years ago. When I connect dots backward, I find myself to have achieved something for which I never thought to have enough courage and strength. Among His countless blessings, yet once again He renders me to believe nothing is impossible – to believe in dreaming, day dreaming and to believe in 'sky is the limit'. All praises and magnificence to Him for awarding me the strength and nerve to complete this thesis.

My sincere gratitude and respect to my supervisor Dr. Zahid Anwar for the continuous support of my Masters study and research, for his patience, motivation, enthusiasm, immense knowledge, invaluable help, constructive comments and suggestions. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my post graduate studies.

I would like to thank my thesis committee: Dr. Ijaz A. Qureshi, Dr. Hamid Mukhtar and Dr. Adnan K. Kiani for their encouragement, insightful comments, questions and suggestions. I am also indebted to my teachers throughout my life who enabled me to cherry-pick conscience through both their speech and silence and imparted highest standards of values, principles and essence in my character. I would also express appreciation for SEECS Administration, especially Mr. Abid for guidance about research phase official processes and enabling me to meet deadlines.

The support, trust and understanding I received from my Family escorted me to the competition of my degree. My Mother and Father have always been great source of inspiration for hard work, dedication and sincerity to one's tasks and responsibilities. They kept my hopes and motivation high during hard times and empowered me to believe in better and success. It isn't only me, infact, no one would be able to pay back thanks to their parents – ever; in any form in this world. I fall short of words to pay thanks to my caring sisters who have always guided, assisted and motivated me on every task and challenge I undertook in my life and always incented me to strive towards my goals.

I earned immense support, assistance and motivation from my friends. I thank all my friends and fellows for lending support in completing this thesis, especially Mohammad Ali for sharing the vision of my research, the stimulating discussions, for the sleepless nights we were working on our tasks together before deadlines, and for all the fun we have had in past years.

I cannot forget to pay thanks to my employers A.S. Enterprises and Tele Taleem (Pvt.) Ltd. for understanding my tough routines, helping my schedules and supporting me during the research.

– Humayun Ali

Table of Contents

List of Figures

List of Tables

Abstract

Smartphone applications are growing in their popularity, this has also brought them under greater threat due to growing smartphone malware attacks. All platforms for smartphones have faced exponential growth of malware in recent years. Bright future of open source platforms are augmenting this figure to grow phenomenally as they continue to capture huge market share. Moreover, smartphone malware are targeting anything from communications, location, personal or identifiable information and are growing to be very sophisticated in nature. A number of detection and defense mechanisms have emerged in the last decade to tackle the mobile malware phenomenon but alarmingly they are mostly ineffective. It has become extremely important to arrange this huge influx of information better understand the smartphone malware problem.

A taxonomy of smartphone malware has been proposed on the basis of their different dimensions. Dimension means to view the smartphone malware from different aspects. Various aspects include; how malware attack the system, their spread or propagation mechanisms, circumvention techniques they adopt to evade smartphone security, architectural layer of smartphone platform they attack, etc. The proposed taxonomy classifies and categorizes both kinds of malware, those found in the wild as well as those which are proof of concept.

Severity of smartphone malware is also discussed for those found in the wild; based on user concerns. Severity is in the terms of the level of threat user feels while keeping in mind the loss it can cause if a malware succeeds to achieve its aim. The severity of malware is presented with respect to user concerns in a graphical manner based on the samples of malware families that are collected from the wild and through various sources.

Chapter 1

INTRODUCTION

Innovation is hard. It really is. Because most people don't get it. Remember, the automobile, the airplane, the telephone, these were all considered toys at their introduction because they had no constituency. They were too new.

– Nolan Bushnell

1. INTRODUCTION

Recent years have witnessed a massive growth in smartphone sales and its consumer base. For the very first time in 2011 (particularly Quarter 4) smartphone sales surpassed that of PC which constitutes notebooks, tablets, and desktops [1]. This development was a result of the revolution in computation and communication brought by the smartphones. Smartphones have enabled their users to stay connected everywhere and all the time through emailing, social networking, file sharing, mobile banking, gaming, audio/video capturing etc. The revolution brought with it a number of IT industry giants such as Apple, Google, Microsoft, Nokia, Samsung, RIM, etc. who are competing with one another in terms of both the smartphones hardware (features such as touch screen, camera, processing power.) and software variants (operating systems, applications, application markets and their usages.). Each of these vendors have captured their own market shares. According to a recent study by Gartner, with respect to the OS, Android (Google) leads the smartphone sales up-to the second quarter of 2013 with almost 80% of the market share; iOS (Apple) holds 14.2%, Microsoft Windows (Nokia, etc.) 3.3% and Blackberry (RIM) 2.7%. Symbian OS (Nokia, etc.) for smartphones has reached its lowest of 0.3%, surpassed by Bada OS (other non-branded) having 0.4% [2]. The tremendous increase in smartphone usage has been observed since 2011 and does not seem to be declining in the near future.

The darker side of the smartphones sales growth picture is the equally expanding incursion of the smartphone malware. Towards the start of 2013, McAfee Labs [3] established that with the increase in the smartphone usage; spread of malware will increase as well. They counted around 37k malware samples [4], which is manifold swollen to the counts received in 2011. While computing security providers are crunching these numbers in terms of their market growth above $86 Billion by 2016 [5], black and grey hat communities are also emulating these numbers aiming to exploit this extended attack surface, specifically for smartphones shipped with open model operating systems. According to an estimate, by 2015, 60% of employees working in enterprises having interaction with information systems would be switching to smartphones for their business activities [6]; which will make it a serious concern for enterprises. The problem can aggravate if enterprises adapt the “Bring Your Own Device” policy [7] for their employees; opening new doors for phishers to break into organization’s network for various illegitimate objectives.

Smartphone malware is not a new concept, nor are the concerns and efforts by grey/black hat communities. In fact it is prevailing since the beginning of 21st century [8]. One of the worth mentioning efforts was from Blitz Force Massada; a security research group, who were the first to attack Android in late 2008 as proof of concept [9]. The experiment proved that smartphones were vulnerable to certain types of attacks, which was successively proved in later attempts on various other platforms as well. Smartphone malware have been growing gradually and some of them; both proof of concept and those found in wild, are highly sophisticated. Soundcomber [10], iKee.B [11] and Obad [12] are some of those highly sophisticated malware which specifically target the smartphone users' privacy and phone control.

It seems that the smartphone platforms would remain more vulnerable compared to their PC based counter parts, as the growing threats include all kinds of malware such as mobile backdoors, exploits, spywares, Trojans, rootkits, ransomeware kits, bots etc. These viciously high count malware have uncapping abilities like stealing sensitive information (e.g. contact books, passwords, text messages, multimedia, GPS Coordinates, Phone Identifiers, credit card and PIN numbers), making calls and text messages to premium numbers, click frauds, remotely controlling the device, rooting devices without owner's consent [4], holding mobile devices hostage, installing apps from markets without user's concerns, corrupting user data and many more. Year 2013 has also witnessed the largest online mobile-money theft of around $5,700 of bitcoins via exploiting a mobile operating system's component. It was due to a flaw in the Android Operating System, confirmed by Google [13], which involves a critical weakness i.e. inability to generate strong pseudo random numbers for bitcoin exchange. The bitcoin attack was performed on Android OS, leaving thousands of applications vulnerable to this exploit. It also establishes the fact that the popularity of a smartphone platform and it's availability with a majority of smartphone vendors in all price ranges; is directly proportional to the amount of consumers brought into the circle of potential targets for malware.

1.1. Motivation

The grey-hat and research communities have jumped into the competition of identifying the weaknesses and vulnerabilities in smartphone operating systems. The trend in this case has always been to observe the platforms having relatively open model because of wide literature

and large consumer base. Since the introduction of first iOS device in 2007 [14] [15], first Android Device in 2008 [16] [17] and first Wi-Fi enabled RIM Blackberry Smartphone in 2007 [18] [19], efforts are being made to define security solutions for their respective operating systems to nullify the security loop holes inherent in them. Different kinds of defense mechanisms including extensions, new modules, code analyzers, code certifiers etc. have been devised by the researchers so far. Just like malware, their defenses also vary in different ways, mostly in terms of detection and analysis techniques and the architectural level at which they reside. A wide range of defenses have been suggested to overcome certain kinds of attacks implemented by Soundcomber [10] Contact Archiver [20], RootKits [21] [22] Single Process Parasites [23] and ikee.B [11] etc.

1.2. Problem Statement

"Methodically studying smartphone malware attacks; to systemize, organize, classify and categorize the available knowledge in the form of a comprehensive taxonomy of malware attacks for major smartphone platforms"

1.3. Contribution

Hence in this contribution, this plenteous knowledge is systemized, organized, classified and categorized along with proposing a comprehensive taxonomy of malware behavioral features and attacks. Little amount of work has been done to classify knowledge about smartphone malware including a few surveys with their focus specifically on either one platform or on particular kind of malware. Approach followed in this thesis also validates itself by correlating the malware attacks against proposed dimensions of malware attack taxonomy.

The findings of research community regarding Android OS Security in particular are a deriving factor of this thesis. Dimensions of the proposed taxonomy are streamlined by multiple well-apprehended proof of concept contributions and malware in the wild attacks on Android OS; while this taxonomy fits well for all popular smartphone platforms.

Chapter 2

BACKGROUND

AND

LITERATURE REVIEW

History is not the story of heroes entirely. It is often the story of cruelty and injustice and shortsightedness. There are monsters, there is evil, there is betrayal. That's why people should read Shakespeare and Dickens as well as history ~~ they will find the best, the worst, the height of noble attainment and the depths of depravity.

– David C McCullough

2. BACKGROUND AND LITERATURE REVIEW

This section highlights the work done in the field of smartphone malware. This thesis is about classification and systemization of knowledge already available about the malware to facilitate further research and development. The overview given in this thesis is with the perspective of attacking mechanism of the malware.

2.1. Analysis of Security Mechanisms in Smartphones

To understand the loopholes exploited by the malware, a critical analysis of the security mechanism of the smartphones was the starting point of the research on malware. Shabtai *et al.* [24] in their research discussed about smartphone security incorporated by Android, the Linux Kernel and the environment in the android smartphones. They identified threats and classified them into five categories or clusters.

First category threats are the ones, which exploit the permissions acquired by certain installed applications to compromise the integrity and confidentiality of the smart phones. Such threats are practical and likely to happen. They can harm the device as well as the private information of the user. The threats due to the weaknesses in the Linux Kernel or the system libraries, which are used by the developers as a part of the SDK fall in the second category of threats. The likelihood of such incidents happening and thus compromising the availability, confidentiality and integrity of the device was proved. According to their research, these threats have less probability but if they do occur, their effect on the device is quite destructive.

Third cluster deals with the threats which compromise the private data stored in SD card or the memory of the device. Such data can have critical information about the individual who owns the device, which may include PIN codes, passwords and credit card numbers etc. Fourth category of threats is the one that exploits the absence of predetermined quota for smartphone applications to use device resources such as RAM, disk storage or main memory etc. Thus any application can attempt to sabotage the CPU affecting the overall performance of the device.

Lastly, the fifth category of threats originates from the vulnerability of connecting a device to other devices through network or port etc. There is a chance that an application, most probably malware, may use the device it is residing on, to affect another device for instance by launching attacks through SMS etc. Shabtai *et al.* [24] also provide recommendations for mitigating and avoiding attacks as mentioned above, but the defense mechanisms are out of the context of this research.

2.2. Attack Mechanisms of Smartphone Malware

The clusters discussed above are generic as far as the mechanism of attack is concerned. Some researchers have gone a step further and classified the attacks being specifics about their methodology. For instance, La Polla *et al.* [25] have found that there are six ways in which a smartphone can be attacked. These methodologies are not specific to the smartphones alone but cannot be ruled out anyway.

La Polla *et al.* [25] states wireless networks to be one of the channel malware can exploit to get to the Smartphones. Smartphones' smartness is depended upon the wireless network but the same make them vulnerable to the attacks as well. The most common method of attacking via wireless network is eavesdropping over the packets during communication, thus stealing information, which is valuable to the user. Not only the Internet but also the Bluetooth networks can be exploited for such purposes.

The next in the line are Break In Attacks, which exploit a weakness of the system files or other programming vulnerabilities such as buffer overflow etc. During this attack the attacker gets control of the device and initiates further attacks on the same device or others connected to it. Example of such an attack is demonstrated by Doomboot .A [25], which installs corrupted binaries having further Trojans, installed in them, to the C:/ drive of the device.

The most complex, effective and broad area of attacks is the infrastructure attacks. Infrastructure is the entity that enables the phone to perform its primary services i.e. sending and receiving texts as SMS as well as making and receiving calls. In smartphones, this

infrastructure further expands towards GPRS and Internet facilities. For example, if an intruder is able to send messages from a device without the consent of the owner while exploiting all the outgoing communication channels, he can block the communication channels thus disabling the user to receive or send even basic calls and SMSs'. La Polla *et al.* [25] has further explained this phenomenon by dividing it into sub categories of GPRS and UMTS.

Next in are the worm based attacks. Worms are the standalone malware which replicate themselves to harm not only the device they reside on but mostly other devices connected to the infected on through the network. La Polla *et al.* [25] further characterize such an attack into three different models based on the transmission channel, the spreading parameter and the user mobility.

In addition to the conventional transmission channels i.e. the calls and texts as well as the GPRS, Bluetooth have recently joined the league of the transmission channels exploited by the malware. The worms especially find the Bluetooth to be of great interest in spreading themselves for infecting different devices.

Not only that the worms use these transmissions channels to infect the devices, they also sometimes infect the network channels themselves. They occupy a larger part of bandwidth thus effecting communications. These effects are not then confined to one infected device only but are spread worldwide wherever the infected network terminates. More alarming is the fact that their spread usually required just one click from the user.

Botnets is also a part of the categories; La Polla [25] has divided the smartphone malware attack into. Botnets, also called Zombie army, are a network of infected computers given the task of spamming the network without the knowledge of the computer owner. Mobile networks, being different in infrastructure than the Internet were considered safe from such forming but, as the phones are becoming smarter by connecting to the Internet, the potential of such connected devices to becoming bots is increasing rapidly. La Polla *et al.* [25] have further

investigated that a command-and-Control network can be formed by exploiting the Bluetooth, SMS, Internet and other channels individually or by creating a hybrid from a smartphone.

Lastly, and most importantly, they discuss that the user driven attacks where there isn't any technicality involved but the user itself overriding security mechanisms. A lot of research is going on to educate the users to avoid such occurrences. They occur mostly when the user take the security measures suggested by the OS or any service provider for granted. When surveyed, more than 50% percent of the smartphone users were not sure if their data was encrypted and thus safe from such attacks. The overriding of security mechanisms usually takes place when a social media hosted application is provided permissions to access private data such as contact list. They also occur through Bluetooth file sharing.

La Polla *et al.* [25] further goes on to investigate into the intentions of the attacker; the possible reasons an attacker would launch an attack. Some of the reasons provided are collecting private and confidential information for monetary benefits such as credit card numbers bank transactions etc. Others are sniffing for sensitive data and the denial of service attack. They further propose defense mechanisms for their investigated threats, which is beyond the scope of this study.

To further verify the security threats and mechanisms of malware to get them to work, researchers; as a part of their researches, performed some proof of concept attacks as well. Soundcomber [10] is one such example where the researchers exploited the sensors installed in the smartphone to steal sensitive to harm the user. Schlegel *et al.* [10] assumed that the application has limited permissions i.e. user does not install an application which requires to access microphone and internet simultaneously. Therefore, Soundcomber is in the form of a package having two separate applications. One being the data collector keeps a lookout for sensitive information by analyzing the sounds from speaker, microphone as well as the touch sensors. It saves the information it reckons to be of any importance and notifies the data transmitter applications. As the name suggests, the data transmitter application has permission to access network, thus it transmits the sensitive data acquired by the data collector to the third party. Soundcomber [10] as a proof of concept, showed how their application was able to

record credit card number by analyzing the spoken as well as the tones which result from touching the keyboard on screen while typing.

As far as the communication between the two applications (Soundcomber – data collector and transmitter), it takes place through covert channels. The covert channels in the smartphone can be Vibration, Volume and Brightness settings. The Vibration channel is specific to Android. Certain applications can change the vibration settings of the phone, once they do, a notification is sent to the interested applications. Soundcomber uses such notifications as a covert channel for communication. However, the Volume settings, once changed, are not broadcasted automatically and the interested applications have to check for themselves. Thus, synchronization is required between the applications. The third and the most interesting covert channel is the touch sensitive screen, which is also called the 'visible invisible channel'. The screen when idle goes dark to save on the power and awakens only by a touch. However some applications, particularly in Android, can request a 'wake_lock' from the system, which prevents the screen from dimming out. This screen awakening and sleeping shoots notifications to the applications, which is used as signals in communication between the applications.

In another similar research by Ali *et al.* [20], the architecture of the Soundcomber was enhanced and a new covert channel was introduced between applications. They used file permissions for communication, which rendered stealing process more complex and a bit fool proof. The data collector, after collecting the data creates a private file and writes its permissions to 10 other files. The permissions include the information the collector wants to give to the deliverer app. The deliverer app awakens as soon as the file is created, it reads the permissions, extracts the useful information, transmits it and deletes the file.

The above mentioned researches about the threats, mechanism of attacks and the proof of concept experiments have been effective individually, yet it is the need of the hour to classify and systemize this information for the sake of further research and refinement of the results.

Chapter 3

METHODOLOGY

No one is an artist unless he carries his picture in his head before painting it, and is sure of his method and composition.

– Claude Monet

3. METHODOLOGY

Since the purpose of the study itself is to provide a better methodology for researchers to device defenses against malware; the thesis could not be completed without getting to the information while following certain path. It is important to outline the research methodology to give future researchers a path to follow. Following is brief description of the phases of this research, as indicated in Figure I:

3.1. Literature Review

To gather knowledge about the extent to which the classification and systemization of the information about malware has been done, the research already been done about the subject under consideration was read through. Not only the text that was related to the systemization itself but the malware attacks was also studied and analyzed. To get an insight about the type and their effects on the smartphones a comprehensive study is inevitable.

3.2. Problem Selection

After a great deal of literature review, the problem which thesis will be addressing was identified. It was in fact the difficulty faced in the literature review itself which served as the motivation to take up the subject for this dissertation. Absence of proper categorization of the information already available is the need of the hour to avoid similar contribution and waste of time in research which has already been done. Also, the categorization can help the scientist to figure out which problem to address in what way.

3.3. Malware/Attacks Instance Collection

Since the direction had been chosen, the next step was to start up on the actual research. Skimming through papers was not enough to attain the objective thus different samples and instances of the malware reported thus far were collected. Smartphones from different vendors operating on various operating systems such as Samsung's Android devices, Apple's iPhones, Blackberry's and Nokia's Symbian operated devices etc. were acquired for analysis. Different applications installed on these devices through third party app stores were studied and how they handled sensitive information was noticed. Also, the complaints that these companies

have been getting about the smartphones security were studied and the causes of discomfort were looked into.

3.4. Getting Detailed Overview of Malware/Attacks

After acquiring the samples the next step was to squeeze out the information relevant to this research. Thus, not only the attacks or the malware were analyzed but the intentions behind launching these attacks were comprehensively studied as well as the state they leave the smartphones in. If the basic purpose of the development of the malware is known not only the already active malware can be stopped but the future attempts which could be made by exploiting the existing loopholes in the system can also be predicted.

3.5. Developing the Taxonomy of Malware/Attacks

The information and statistics gathered from analyzing the malware attack was then passed through the phase of taxonomy. The malware attacks were categorized on the basis of their propagation method, their stealthiest in doing their job, their purpose of attack and the effects their activities have on the host smartphone. Some of the malware fall into more than one categories. The taxonomy is quite comprehensive. Thus the information is quite focused and converged which will help the future researchers to get to their desired information without wasting time into reading the material which overlaps and proves redundant.

3.6. Validating Malware Taxonomy

No research is complete without its verification. The verification of the research involving taxonomy is a quite different than the rest of the fields. It involves a comparison of the taxonomy proposed with the ones already available. It also requires to be futuristic enough to accommodate further research into the topic which is a taxonomy. Therefore, the taxonomy proposed in this thesis about the malware attacks was validated by looking into the taxonomies already done and was found in line with them. Since this research involved predicting vulnerabilities which could be exploited by the malware developed in the future, the taxonomy proposed by this thesis will accommodate the malware which might sprout later.

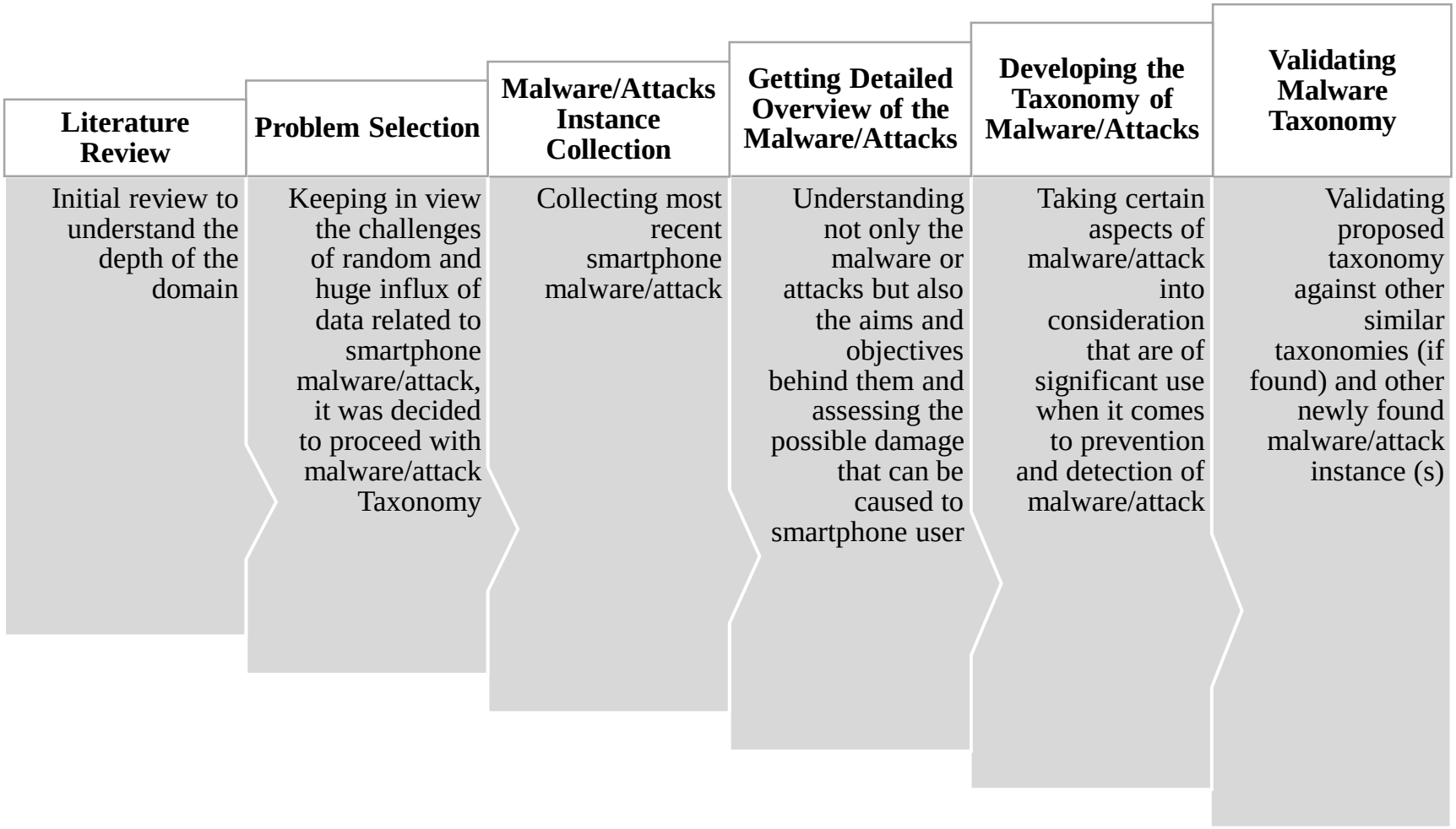

Figure I : Methodology

The important achievement of Apollo was demonstrating that humanity is not forever chained to this planet and our visions go rather further than that and our opportunities are unlimited.

– Neil Armstrong

Chapter 4

TAXONOMY OF SMARTPHONE MALWARE: CLASSFICATION AND CATEGORIZATION

4. TAXONOMY OF SMARTPHONE MALWARE: CLASSFICATION AND CATEGORIZATION

This section proposes a taxonomy of smartphone malware from various dimensions. The word dimension depicts to view the smartphone malware from different perspectives and aspects. The proposed taxonomy classifies and categorizes all sorts of malware; those found in the wild as well as the ones known as proof of concept. The different aspects of malware those were taken under consideration are explained one by one in following subsections. In order to facilitate readers, significance for main dimensions with respect to smartphone malware problem is also included. Also sub dimensions are explained with their brief introduction, in taxonomies to follow

.

4.1. Malware Type

A malware can harm a system (smartphones in this case) in any means as established in the dimension of 'Malware Aim'. There are two pre-requisites for a malware to harm a system; i.e. penetrate by evading security mechanisms and then to sustain long enough to achieve its objective of harming the system. Given this, it is important to dissect the smartphone malware based on the aspects of their penetration, operations and persistence involved throughout their existence in the system. It is interesting to find a malware stronger in one of the three aspects than others. For example Trojans are known to possess effective mode of penetration but may not be as stealth enough as a rootkit to stay undetected in the system. This dimension would also help to understand the relationship between the malware and system under attack. This deduces following three sub-dimensions:

4.1.1. Mode of Delivery

Mode of delivery indicates that how a malware makes its way into the system. It signifies the lack of acumen of the user that may be exploited by a malware via social engineering attacks to infiltrate into the system. Also mode of delivery implies the systems' weaknesses and vulnerabilities. These vulnerabilities could be inherent or related to systems' design [26], application code [27], communication protocols, hardware sensors and services [28] inside the system. Following are types of malware that are significant in terms of their mode of delivery.

4.1.1.1. Trojan

Malware known as Trojan seem legitimate/non-malicious but have a malicious intent. Trojans mostly compromise users' privacy. Usually, popular applications get repackaged as Trojan, for example; an application pretending to be a media player can steal users' data. Schlegel *et al.* [10] created a proof of concept Trojan called Soundcomber to demonstrate the risk of privilege escalation attack using application collusion. Soundcomber extracts and steals sensitive information from user's recorded voice. It uses social engineering trick like masquerading a speech recognition application, to get installed on users' phone. A similar proof of concept malware was demonstrated by Ali *et al.* [20] called Contact Archiver, which also uses social engineering trick by masquerading an archiving application to get onto users' phone. Some other proof of concept Trojans including ZSONE (taken off from market later) were discussed in McAfee Report [9] demonstrated in late 2008. Apart from the proof of concept attacks, FakePlayer was the first SMS Trojan that appeared to be a media player, which used to send SMS Messages to premium numbers without the user realizing [9]. Zhou and Jiang [29] discusses a number of real world Trojans which are potential threats nowadays including; classic video game 'snake' which was able to send GPS coordinates after every 15 minutes to a remote server. 'Walk and Text [30], appearing to be a useful application enabling the user to type, walk and see beyond the phone at the same time, was actually a Trojan and used to steal users' contacts and send out embarrassing text messages.

4.1.1.2. Worm

Worm for smartphones behave differently than worms for PCs or Server Machines. PC based worms normally rely on protocol vulnerability to evade by mainly doing port scanning attempts. But smartphone worm rely on different techniques to evade such as infecting storage cards [31], Bluetooth, and MMS [28]. Cabir was one of the first smartphone worms appeared in the wild targeting Symbian OS [32] by frequently scanning and infecting its peers on the same personal network using Bluetooth and ultra wideband. Mabir.A [33] a variation of Cabir worm is able to spread via MMS. It waits for a SMS or an MMS to arrive and then sends a copy of itself as an MMS reply. Ikee.A [34] Worm is able to deceive the iPhone users into downloading and installing it from a 3rd party app store on a jailbroken device. Merogo SMS worm [35] is a malware that spread through "drive by download" mechanism which was initiated by clicking on malicious link received in an SMS message.

4.1.2. Malware Activity

Malware activity is more of operation related aspect which connotes the variation in attacks performed by different malware; where they work to actually breach, steal, or gain access to information of interest. One of the significance of malware activity is that it reveals the damage and its severity in the situation where it successfully performs its operation after getting inside the system. The severity of damage is subjective and depends on the kind of attack performed. Usually compromise of financial information like credit card number or pin are considered to be more fatal, however, for some users, loss of photos, or location disclosure could be of more concern. Following are the types of malware which are important with respect to their activity are discussed.

4.1.2.1. Spyware

Smartphone spyware are usually developed by making modifications to famous and genuine applications to spy on the users [36]. The modifications are made by malware developer having the access to the source code and binaries of renowned applications. Termed as repackaging attack for Android OS; this attack may be replicated in any language for any smartphone platform. Since it pretends to be legitimate application and generates usual network traffic, therefore; it goes undetected by layman users and trivial anti-virus software. There are certain spywares available in market to spy on almost all famous platforms. FlexiSPY [37] and Mobile Spy [38] are examples of smartphone spyware having rich features including remote listening, SMS and email logging, location tracking, call logging, web URL logging for Android, Symbian, RIM and Windows based smartphones. Mobile Spy is capable of providing spying features on Jail broken iPhones as well, however, FlexiSPY lacks this functionality. FlexiSPY has also been declared/labeled as a malware by F-Secure [39] (formerly Data Fellows [40]; an online security provider for multiple devices and platforms) for more than once in past years. It has sparked a controversy between two companies resulting in exchange of fuming news posts [41] [42] [43] and virus description pages [44] [45] for Symbian S60 platforms [46]. Although these activities are because of professional competitiveness; however, they do raise doubts and hesitation about the genuineness of spying software available in the market, claiming to be legitimate.

SS8 Interceptor [47], termed as a spyware; was developed by Legal Interception Company SS8 [48] and was rolled out by Etisalat [49] through WAP in United Arab Emirates (UAE) for BlackBerry smartphones. It claimed to augment the performance of BlackBerry and a token for the continuation of services. However, it was reported to be draining the battery of smartphones rapidly. Gunasekera [47] analyzed one of the version of this interceptor and found it to be intercepting only outgoing emails from the accounts configured at infected device and not sending any info to the central server. However, other sources [50] claimed that it was not written and planned very well to intercept email and text messages across all UAE BlackBerry smartphones; making it a large scale deployment which brought the central server down, causing BlackBerry devices to make repeated request which resulted in quick drainage of battery. If it would have been planned well, it would not have let any upgraded smartphone to experience such rapid battery drainage and slow downs. Kakao Talk [51], which is a smartphone messaging application in its compromised version was a spyware noticed by Citizen Lab; allegedly designed for political cyber-harassment & surveillance against Tibetan and Uyghur activists. It was purposed to surveil messaging history, location on cellular network and contact lists. Templeman *et al.* [52] demonstrated the idea of virtual theft by developing and evaluating a visual malware named PlaceRaider that is able to capture images of smartphone user's physical environment using smartphone's camera. The proof of concept malware then generates a 3D view of the user's environment where an attacker can steal virtually important information like information written on finance related documents, computer screen, and other personal information from the generated image.

4.1.2.2. Bot

Bots serve the purpose of using the smartphone as a relay to perform specific kind of malicious activity in collaboration with similar (compromised) smartphone hosts. Bots are mostly used for bulk attacks on remote hosts/servers, usually DDOS. Smartphones from all over the world are becoming part of the smartphone botnets. These bots are mostly used to infect other devices, generate spam, etc. Infections are mostly injected through free popular games [9] with the help of social engineering techniques. Once these devices are infected, they get in contact with a command and control server for instructions. They become a part of a botnet permanently. Instances of smartphone bots were noticed by a security firm, known as Dambala [53] , in the beginning of 2011.

Porras *et al.* [11] analyzed iKee.B [54] [55]; an iPhone bot registered in late 2009. It exploited vulnerability in SSH, allowing a default user 'alpine' to login. Using this vulnerability the bot used to propagate and run its installation script on any iPhone. The installation script downloaded iKee.B code, installed it on the infected device and set it to run on boot time. iKee.B was able to contact its command and control server to receive certain instructions. Xuxian's team discovered a malware in third party applications markets and known as DroidLive [56]. DroidLive disguised as a Google Library and attempting to install itself as smartphone's admin application; but actually possessed features as those of a bot. It connected to its command and control center to engage in activities of texting, calling on premium numbers and collecting personal information.

4.1.3. Stealthiness

Mobile malware can also be classified on the basis of their stealth behavior. Malware discussed here are noteworthy in their aspect of keeping their operations unnoticed or staying undetected or hidden by adopting a low profile or changing normal system's control flows to deceive a profiler or an anti-malware. Stealthiness poses challenge to built-in security or 3rd party anti-malware solutions to detect malware having stealthy behavior. Detection becomes difficult due to the fact that a malware could be running as privileged application or may manipulate system normal flows/operations to perform its activity.

4.1.3.1. Rootkit

Rootkits simply hide themselves from being detected to perform various malicious activities with root/admin rights on their hosts. Rootkits are hard to detect and remove from an infected system. Smartphone platforms can be a potential ground for rootkits. Infecting a smartphone by rootkit is about finding exploits and then hooking the device [21]. Papathanasiou and Percoco [57] developed Mindtrick, a malware that operated on kernel level. This rootkit used to get activated on making call to specific numbers in order to transmit a reverse TCP over Wi-Fi/3G shell to the attacker. It was a proof of concept rootkit and no examples of such kind of malware have been found in the wild so far. Bickford *et al.* [58] demonstrated three possible rootkit attacks with serious consequences where they spy on GSM conversation, breach privacy via tracking GPS location and exhaust battery for DoS. They also predict that due to

the popularity and growing market of smartphones, attackers will soon begin deploying rootkit attacks for malicious objectives. Xuxian Jiang research team [59] identified vulnerability in various platforms that could be exploited by clickjacking rootkits, which may download along with an application and manipulate the device. The manipulation include phishing on user data by disguising as legitimate application or taking full control of device. Same team also report two dreadful rootkits – RootSmart [60] and GingerMaster [61] that exploit with the help of Android rooting image known as GingerBreak, launched by XDA Developers [62]. Both the rootkits hide in the system and perform their malicious activities in response to various system level broadcasts and events those they have registered for. GingerMaster embeds/repackages the root exploit with itself while RootSmart downloads the exploit after settling into the system to escalate its privileges. Both the malware later download other kind of malware to the system from their command and control server. DroidLive [56] is one instance of such malware which were downloaded later by RootSmart. DroidLive also received instruction from command and control server for texting, calling on premium numbers, and privacy breach.

4.2. Aim of Malware

As discussed in section 4.1.2, a malware can be categorized in terms of its type of activity/operation which ultimately augments a malware's ability to attain some objective, which can be termed as 'aim of malware'. This dimension links various activities of malware with their aims. This also provides an insight into what type of malware usually cause what type of damage to a smartphone user as also supported in section 6. Malware's primary aim is to get to its target by performing the intended malicious activity. Target of malware indicates the piece of information they are trying to grab by evading security mechanisms. The target could also be to take control of the user device; or causing financial/data loss to the user. Providing unsolicited information is also generally classified as a form of attack. Mostly, spam and annoying ads are counted in this category. Following sub-sections elaborate the further categories of this dimension.

4.2.1. User's Privacy Breach

So far user's privacy has remained the most attacked and misused targets of malware. Most of the smartphone malware families target user's privacy and are very difficult to track. Privacy and the impact it may cause in case of loss and breach, varies from user to user. The privacy-sensitive data is mostly comprised of SMS Messages, Contact Numbers, PIN/Credit Card Numbers, User IDs, Passwords, Images, Videos, GPS Coordinates, IMEI (hardware credentials), user's web traffic, etc. An instance of a serious malware that affected Android Phone users was an instance of Walk and Text [9]. It used to forward SMSs to all the contacts persuading them to download a package with heavy charges for the SMSs. It also used to collect user's personal information like IMEI number etc. to send it to a remote server. Zhou and Jiang [63] discussed two critical vulnerabilities in content provider and Android application i.e. passive content leakage and content pollution. Passive content leakage takes place when an application is unable to protect its (internal) data properly. This application may unintentionally allow other applications to access its data. Content pollution is the phenomena where an application is allowed to access or to manipulate other application's data with proper authorization or permission set.

Smith [64] discusses how iPhone's UDIDs can be misused and the repercussions which follow. The intended use of the iPhone's UDID is for various personalization settings and certain application needs e.g. game's highest score etc. Apple has allowed the use of UDIDs by any application to personalize settings for the device. Things get complicated when this UDID is used to track user or user account (user preferences) especially on the Internet and associate certain other information that can further be shared with a third party.

A report [65] released by Research In Motion Limited (RIM) [66], now known as BlackBerry Limited [67], mentioned MAPI Attack. Attack is initiated when a malicious or benign user uses a malware from inside of an organization's corporate network. If the attack is successful, it substitutes a benign RIM Based Blackberry device with malicious RIM Based Blackberry device on a company's network where BlackBerry Enterprise Server is deployed. As a result; the malicious device belonging to the attacker is able to collect the privacy sensitive information belonging to the benign user's BlackBerry device. This illegitimate access of

information includes emails and contacting organization's internal servers on the corporate network.

4.2.2. Gaining Control of the Smartphone

Gaining control of the device has been one of the most popular intent of malware. Sending SMS and making call to premium numbers are common instances of this sub-dimension [9]. It may cost users a heavy amount resulting from unintended SMS and calls. Merogo SMS worm [36] is one such malware which spreads through "drive by download" mechanism; by clicking on a link sent over in an SMS. It was only found in China and had Chinese language characters in the text message. As expected, it sent out text messages to premium numbers, causing financial loss to the user of infected smartphone [68]. Gaining control of the smartphone also includes the instances such as installation of update and download of files or, draining smartphone battery or, if the device is under influence of a command and control server (under a botnet), all without user's knowledge and consent. iPhone's SMS Fuzzing [69] is a type of root exploit that used to take control of the device by exploiting a vulnerability with the help of malformed SMS messages. It was fixed by iPhone later.

4.2.3. Unsolicited Information / SPAM

Free applications mostly come with advertisement banners that cover a part of the screen. Free applications allow both the developer and Advertisement Company to benefit financially from users' clicks. The frequent display of advertisement banners becomes annoying when the user accidently clicks on them, taking him to another view e.g. browser or application market, every now and then. Shekhar *et al.* [70] discusses click-frauds in which a malware developer generates fake clicks on the ads being shown on free applications' banner area to earn extra money. Stream of irrelevant SMSs from mobile service providers, advertisement companies and social networks such a Twitter, Facebook etc. is also a form of unsolicited spam [71]. These types are mainly from the perspective of usability of smartphone users.

4.2.4. Smishing (SMS/MMS Phishing)

Recently another kind of smartphone attack: Smishing (SMS Phishing) [72] [73] arrived in the wild. Smishing is an attack whereby the user receives SMS text from a number claiming to be an authorized party. The text contains a link to some webpage seeking the user to click on it. As soon as the user clicks, it leads the browser towards a malicious page infecting the device via drive by download, buffer overflow or by exploiting some other vulnerability present in the browser through malicious code. Most of the time, these scams persuade the text message recipient to visit some website or make a call to a number, where the victim is tempted to provide privacy sensitive data including credit card details, PINs or passwords. Symbian Sexy Space [74] pretends to be a legitimate application for Symbian devices but in actuality steals consumers' phone and network information. It also tends to spread via spam text messages to the contacts in a compromised smartphone.

Xuxian Jiang identified another aspect of Smishing attack on android devices [75]. They reported that there is vulnerability present in all Android Platforms which allows any application to generate fake incoming SMS or MMS claiming to be from authorized sender and persuades the user to click on a provided link. One of the alarming facts about such an attack is that it does not require the malicious application to request any explicit permission of WRITE_SMS for launching the attack. This vulnerability is termed as capability leak [76] where hazardous privileged permissions are available to be used by different applications, and do not need to be requested in the code explicitly for the actual use at all. This SMISHING vulnerability has been fixed by Google Android Security Team in Ice Cream Sandwich 4.2 release.

4.2.5. Theft/Fraud

Theft and fraud have been among the cardinal aims of malware from the beginning. They are mostly intended for monetary benefits extracted out from the victim. They also include malware stealing privacy sensitive information from and about user which may be used later to harm user in different ways; but must not be confused with Privacy Breach (as explained in section 4.2.1.) because harming user may also include using stolen credit card and pin numbers to cause financial loss (e.g. Soundcomber [10]) or exploiting users' personal data for ransom

related illegitimate purposes. Moreover, privacy leakages may not necessarily lead to theft/fraud. Malware now also have evolved to target advertising companies where by a malicious application that comes with Ads can generate fake clicks on the ads and earn revenue from each click (known as click fraud). It is possible because the applications that come with Ads (mostly free) normally run as single applications. Ad companies cannot discriminate between a click that is generated by the user and the one that has been automated by some malware. As a result, Ad Companies cannot detect click frauds done by malicious applications [70] [77]. Hornyack *et al.* [78] [79] discuss scenarios where an application may misuse private data within the phone or send it to a remote server and classify similar attacks as privacy risks. Templeman *et al.* [80] Demonstrated the scenario of virtual attack or theft by using smartphone sensors such as camera, mic etc. They discuss how a virtual malware can be a useful tool to capture or gain knowledge about the physical surroundings of a victim including building structure, documents, computer screens etc. This information can be used to carry out the actual theft later. Mylonas *et al.* [81] did a comparative study on feasibility and ease of malware development with the help of rich and extended tools and SDKs provided by all renowned Smartphone Platforms including Android, Apple iOS, BlackBerry, Symbian and Windows Mobile. They conclude that a developer having an average skillset is able to develop a malware with the available development facilities. They demonstrated a proof of concept malware which records and shares with remote server; the users' GPS Coordinates for tracking purposes. The demonstration of attack on all mentioned platforms was successfully implemented, which concluded that these state of the art platforms do not provide sufficient security assurance to prevent one of the major privacy breach done in the form of location tracking attack.

Storm8 [82] (a leading iOS game developer having millions of downloads on their account) was accused when a lawsuit was filed against them for harvesting/stealing information including phone numbers of users which were sold for spamming and other purposes [69]. It was found that almost all of the games by Storm8 tend to capture and transmit the mobile numbers to remote servers [83]. In response to this alert, Storm8 team apologized, explaining that it was a bug left unremoved during development and testing phases and will be fixed in the next update.

4.3. Architectural Layer of Attack

Malware can attack at different architectural layers of the smartphone platforms. There are two main architectural layers, i.e. Application Layer and Kernel. It is the Application layer, which is attacked mostly, however, some instances of attacks launched on Kernel have also been observed and are explained in following sub-sections along with their nature and examples. The architectural layer under attack of a malware defines its demand of privileges to perform its operation. Normally it is seen that malware targeting kernel layers require more privileges. It is also significant in terms that detection techniques vary for malware residing at different layers:

4.3.1. Application Layer Attack

This dimension focusses on malware that infiltrate as an application. Security firms and research groups have noticed a tremendous increase in application layer attacks by new families and samples of malware; McAfee, specifically, has reported it in all their threat reports since the beginning of year 2012 [84] [85] [86] [87] [88] [89]. Application layer is threatened when a user installs an application through social engineering attack designed by the malware developer or, if the device is physically compromised. After the malicious application gets installed on the device; it can be activated by any event including device boot. The damage can be performed either by one application (repackaged app or pure malware) or by multiple applications collaborating to compromise the device's security (collusion attack, privilege escalation etc.). The procedure of attack (by one or multiple applications) arriving at application layer makes another dimension of the taxonomy and is explained in next section.

Applications written with the positive intentions may contain security vulnerabilities, which can be exploited by other malicious applications installed on the same device. Schrittwieser *et al.* [90] Discusses some messaging applications that can possibly be exploited at certain level of their operation (messaging, call initiation etc.) and risk the privacy attributes and sometimes control of the user's phone too. Luo *et al.* [91] discussed the possibility of a malicious application which can steal sensitive data, compromise the integrity of web page and user interaction while it incorporates the web view control for all famous platforms including Windows Operating System for smartphone.

4.3.2. Kernel Layer Attack

Malware in this category tend to manipulate the kernel layer of smartphone platform. The most prominent Kernel Layer Attack found in the literature of smartphone security is rootkit. Major task in rootkit deployment is finding the sys_call_table. sys_call_table contains addresses to kernel's system calls. After the sys_call_table has been located; the address of a particular system call is replaced with the address of desired routine/system call which is termed as hooking. These hooking techniques are discussed at [21] while prominent of them are hooking through /dev/kmem access, by changing vector_swi handler routine or branch instruction offset. As discussed earlier; basic purpose of a root-kit is to hide itself, which is usually done through hooking. Once the hooking has successfully occurred, the desired routine/system call is invoked at a certain point of time where it is executed by kernel under its root privileges. This desired system call running as root can contain malicious code, which can harm the user as discussed in section 4.1.3. A rootkit can infect a device using various exploitation techniques such as control flow attacks. Birsan [92] discussed how TrustZone-based-rootkit can exploit Trusted Execution Environments on ARM Based Devices to compromise communication, data, credentials and money on the device.

4.4. Circumvention Technique (Attack Technique)

Circumvention techniques are those which are adopted by malware to exploit vulnerabilities and evade existing security mechanisms to attack at different layers of smartphone platform. This dimension reveals the weaknesses in the security mechanisms or components of an operating system. Some weaknesses can also be generated by developer due to poor programming practices and granting benign applications with unused permissions. This section discusses some of the most used/discussed circumvention techniques.

4.4.1. Privilege Escalation

Privilege escalation attack is one where a malicious application performs a privileged task without having the necessary permission it is supposed to have. For example, in the case of Android, it allows an application to request some other application to perform a privileged task. This easy-to-use technique is frequently exploited to circumvent Android Permission Model and Reference Monitor.

An advisory was released by RIM Blackberry [93] intended only for Z10 smartphone users and IT Administrators who are responsible for deploying BlackBerry in enterprise environment. They revealed the presence of a privilege escalation vulnerability in a component of BlackBerry Protect [94], which may be exploited by a malicious application if BlackBerry protect is enabled and user attempts to reset their password. Given that all these conditions are met, if an attacker gets physical access of the compromised smartphone; he can log in with the password recently reset by the user. Even if attacker doesn't have physical access, he can still utilize the attained password over Wi-Fi File Sharing and on other places where this password is applicable [95]. There are different variations of privilege escalation attacks out of which confused deputy attack and application collusion attack are the most common.

4.4.1.1. Application Collusion Attack

Application collusion attack is a variation of privilege escalation attack where two malicious applications use their permission sets implicitly to circumvent smartphone security. This attack mostly goes undetected because of its implicit flow. Soundcomber [10] carries out a sophisticated application collusion attack (as proof of concept) on Android; stealing sensitive user information like pin codes, credit card numbers, etc. Soundcomber performs application collusion attack by using two applications – Soundcomber itself; having permission to record voice, and some other application that has networking permission as a deliverer. These two applications use collusion attack while having a collective permission of microphone and network. Soundcomber extracts sensitive information from recorded voice and sends it to the deliverer application, which sends it to a third party server. To circumvent Android reference monitor, Soundcomber uses covert communication rather than the mostly-used IPC techniques. Using implicit flow of information, Soundcomber is able to send extracted sensitive information bit by bit to the deliverer application, while manipulating content managers, file locking, screen brightness and volume features. Contact Archiver [20] inspired from Soundcomber, has enhanced stealth and sophistication by devising another covert communication channel. It steals user's contacts and using the same architecture as that of Soundcomber; sends all the contacts to a third party server.

In addition to the aforementioned techniques, application collusion attack can be carried out in certain other ways as well. Dietz *et al.* [77] show that in some distinctive scenarios, only one application can also carry out application collusion attack. One application with access to privacy information can send the information outside with the help of malicious script/extension that can be run from/with-in a browser. XManDroid [96] discusses how, in an application collusion attack, applications bypass Android's reference monitor by using lower level implicit flow e.g. file system etc. As a proof of concept, Shabtai *et al.* [97] developed a set of applications which exploit Shared-user-ID (applications by the same developer) to carry out an attack similar to application collusion attack.

4.4.1.2. Confused Deputy Attack

Confused deputy attack is a variation of privilege escalation attack where a non-malicious application's (known as confused deputy) permission is used for a malicious activity e.g. downloading a malicious payload, redirecting to a malicious website, etc. Like various smartphone's permission based security model, each application in Android has to request a set of permissions to access required resources on the device. User has to grant the permissions requested by the application at install time. Similarly, malicious applications are also able to request permission to critical resources of the device.

Felt *et al.* [98] discusses some scenarios where a browser having permission to critical APIs (like network access, GPS etc.) becomes a confused deputy when a malicious application (having less permissions) requests the browser to fulfill some task. As a matter of fact, in Android, applications can request operating system components to fulfill any functionality by; running a service (synchronous call), launching an activity using intents [99] [100], sending broadcasts or accessing content providers e.g. Contact Manager etc. In this scenario, a malicious application (less privileged) may request a benign (privileged) application to perform a hazardous task which can infect the device or compromise user privacy.

XManDroid [96] discussed Android's weaknesses (that make confused deputy attack possible) from a different perspective and claimed application-dependent policy enforcement

mechanism to be one of the causes behind confused deputy attack. Another factor that makes confused deputy attack possible could be over-privileged applications i.e. applications with un-necessary permissions discussed in stowaway [101]. The ignorance of the developer about the exploitations possible through particular permissions might be the reason behind the provision of un-necessary privileges to an application.

4.4.2. Exploit or Control Flow Attack

Exploit or control flow attack is a technique which enables a malicious piece of code to execute itself with the rights of the applications/service under attack. Conventionally it involves injecting the malicious payload in the memory space of the application. After gaining access it may download and install a rootkit or a service that may leak the privacy information, and so on; which is difficult to detect and remove. With increase in the number of applications in the market; new vulnerabilities are being introduced/found which can be exploited by malicious applications. Davi *et al.* [102] identified some variations of control flow attacks such as code injection and code reuse attack and their causes. Such as in the case of iOS, where applications are developed in Objective C. Since Objective C is not a type safe programming language, iOS is a major target for control flow attacks, as compared to Android where mostly applications are developed in Java, which is a type safe language Type safety prevents type errors that may normally lead to attacks like buffer overflow. Android also facilitates developers to write applications in C/C++ using native development kit (NDK) [103], which is one of the foremost cause of control flow attacks in such applications as unsafe code in C/C++ which might introduce vulnerabilities [104].

4.5. Feature Exploit

Feature exploit means the misuse of any features provided by smartphone vendors both in the terms of software and hardware. These features include operating system's and hardware component's mechanisms for various application to applications and application to user interactions. This dimension highlights features of smartphone, either hardware or software that are benign by their nature and solely intended for smartphone operations and user facilitation but can possibly be used for illegitimate purposes.

4.5.1. Inter-Component-Message-Passing Misuse

Applications rely on various inter-process-communication techniques for their operation in a shared environment. In case of Android, broadcasts are important feature for applications to communicate with each other and perform their tasks. The broadcast feature of Android allows applications to send and receive broadcasts in a system wide manner. Two modes of broadcasts are available in Android; un-ordered and ordered. Ordered mode allows the broadcasts to be sent to each registered receiver one by one (the order is governed by the priority attribute associated with each registered receiver in their manifest file). In this mode; a receiver can deprive a low priority receiver of the broadcast by terminating it. ACTION_NEW_OUTGOING_CALL is an example of ordered broadcast [105]. If two benign applications are communicating through broadcast where other applications don't know or need to register for receiving the broadcast, a malicious application can receive this broadcast and data transmitted can be compromised. Moreover, if it is an ordered broadcast; malicious applications can abort the broadcast and re-generate the original broadcast with false data; bringing sensitive data into account.

Comdroid [106] discusses some techniques that malicious applications can use to steal implicit intents and read its content if the intent is not protected by any permission. To make that happen; all a malicious application needs to do is to declare an intent filter for the intent type. The authors discussed the possibility of the attacks that may occur as a result of intent misuse. Broadcast misuse occurs when a malicious application receives the broadcast and uses its content for eavesdropping between two applications. A malicious application can also perform denial of service attack by preventing ordered broadcasts from reaching other applications [107]. Another serious outcome of intent misuse is activity hijacking where a malicious activity instead of the actual activity is launched. Similar attack can be carried out in case of service [108]. Similarly intents can be spoofed as well, resulting in a more harmful attack against unprotected intents [109]. Possible threats include; injecting malicious broadcast and launching malicious activity/service [110].

4.5.2. Sensor Exploitation

Sensors exploitation denotes the misuse of sensors and other hardware features provided on smartphone devices including microphone, camera, touch screen, accelerometer, Bluetooth, Wi-Fi etc. Smartphones have witnessed some state of the art sensory malware in the recent past. The proof of concept efforts are the most sophisticated and worth noticing among them. Soundcomber [10] can also be termed as a sensory malware that captures sound (including typed credit card numbers and pins) through microphone at specific intervals and events. Information from the recorded sound is then recovered through frequency analysis and sent to the remote server over the network. This instance of misuse of sensors on a device also invokes the possibility of exploiting other reasonably sensitive sensors including camera, GPS, gyro and accelerometer as well. Marquardt *et al.* [111] emphasize that access to a lot of imperative sensors has been regulated in various operating systems over the past few years [112], however, there still remains a room for sensory malware to exploit other sensors in their unintended ways to infringe user privacy. Marquardt *et al.* [111] have demonstrated an attack through a proof of concept malware by recovering key strokes generated by a nearby computer key-board. Elementary employment of accelerometer sensor makes this malware non-trivial. Recovery of key-strokes is performed with the deployment of least involved equipment as compared to past contributions [113] [114]. They developed an infrastructure of neural network which records the accelerometer data and then analyzes pairs of keystrokes to recover information with an accuracy of 80%. Philip *et al.* demonstrated that on iPhone, the information from the accelerometer sensor captures the emanations from a nearby keyboard, which can be processed by characterizing the key-press vibrations to extract the text being typed.

Cabir [115] as detected by Kaspersky Labs is one of the first smartphones worms that used to spread through Bluetooth by scanning nearby devices. It sent malicious .sis file to the first device found and then on every boot. As soon as malicious file is received, Cabir starts its operation of scanning and sending to further victims. There are also growing risks of keyboard applications for smartphones turning into key logging malware [116]; as noted that SwiftKey, one of famous keyboard application is found as repackaged with key logging malware [117].

4.6. Manufacturer Fault

This dimension encompasses the type of malware which exploit the vulnerability introduced as a consequence of a defect left un-intentionally by the manufacturer of the smartphone during manufacturing, image loading or patch cycle issuing process [118].

4.6.1. Capability Leaks

Woodpecker [76] analyzed a number of popular android devices available in the market and found that the images of Android loaded on these devices by their manufacturers do not enforce the permission model of Android properly. As a result, malware are able to access a resource for which they have not requested the permission in their manifest.xml file. Such consequences are collectively known as capability leaks.

4.6.2. Delayed Patch Cycle

Google has introduced Open Handset Alliance where they have made the base open source operating system available. It can be acquired by device manufacturers and service carriers for customization according to their requirements, capabilities of their devices and the plans they offer (in case of service carriers). With this kind of operating system distribution model, when a vulnerability is discovered, the patch is issued by Google first. Afterwards, it is the manufacturer's responsibility to issue the patch of the exposed vulnerability for their respective smartphone hardware. User is then able to apply the patch issued by the manufacturer [119]. In this patch cycle, if Google finds it necessary to release the patch, it usually takes around four months, but it takes longer than that to patch the vulnerability on users' device. It is because the released patch has to be updated by the manufacturers to suit their hardware which depends upon their budgets and resources available; required by the update. Sometimes, due to the resource limitations faced by the manufacturers; users don't get the updated patch release for their devices at all. If the update is made available by manufacturers and service carriers, it is often of no use since the user's hardware is already exploited due to the long exposure to the vulnerability because patch is already available to explore possible cases to exploit subject vulnerability It enables the malware to exploit the vulnerability in the devices either having delayed patch cycle or those devices which have not received the official release at all. One of the prominent example of such delay in patching

device on user end is of Sony Ericsson Xperia X10 where Google released a patch for Android 2.1 in January 2010 [120], which was rolled by Sony Ericsson in October 2010 for Xperia X10 [121], even after Google had released Android 2.2.

4.7. Malware Activation

After malware intrudes the device using any method of propagation, they have to be activated to launch the attack. Activation can take place either by any system level event or through users' interaction/event with the device. System event activation happens when the intruded malware gets activated through any system generated events, for example inter-process-communication, battery levels, system settings, Wi-Fi/Bluetooth status, system boot etc. In the case of Android, malware may have registered in its manifest.xml file for various intents or broadcasts, on the reception of which it may get activated. The intents or broadcasts may belong to any system level event including outgoing call, SMS reception, battery level notification, system boot etc. Zhou and Jiang [29] mentioned that AnserverBot is activated when either the device is booted, an SMS is sent/received or a network connection is made etc. sp(iPhone) [111] is a sensory malware that decodes and recovers keystrokes from accelerometer readings when it is placed in range of a keyboard under use for typing purposes. sp(iPhone) [111] does not record accelerometer data in a continuously. It performs it in distinct events like on reception of text messages or by regularly probing the activity.

Malware those require user interaction to get activated fall in the category of user event activation. FakePlayer [122] is activated when user clicks on its icon in the applications list. Soundcomber [10] activates itself when the user enters a credit card or pin number after making an outgoing call to a sensitive (banking IVR) number. Walk and Text [9] gets activated when users run it to walk and look beyond their smartphone while writing a text message. sp(iPhone)'s activation can also be termed as partially based on user events because it depends on the user to place the smartphone near a keyboard under use in order to generate readings on the accelerometer. It is also important to note that a malware may never activate if the required event is not performed.

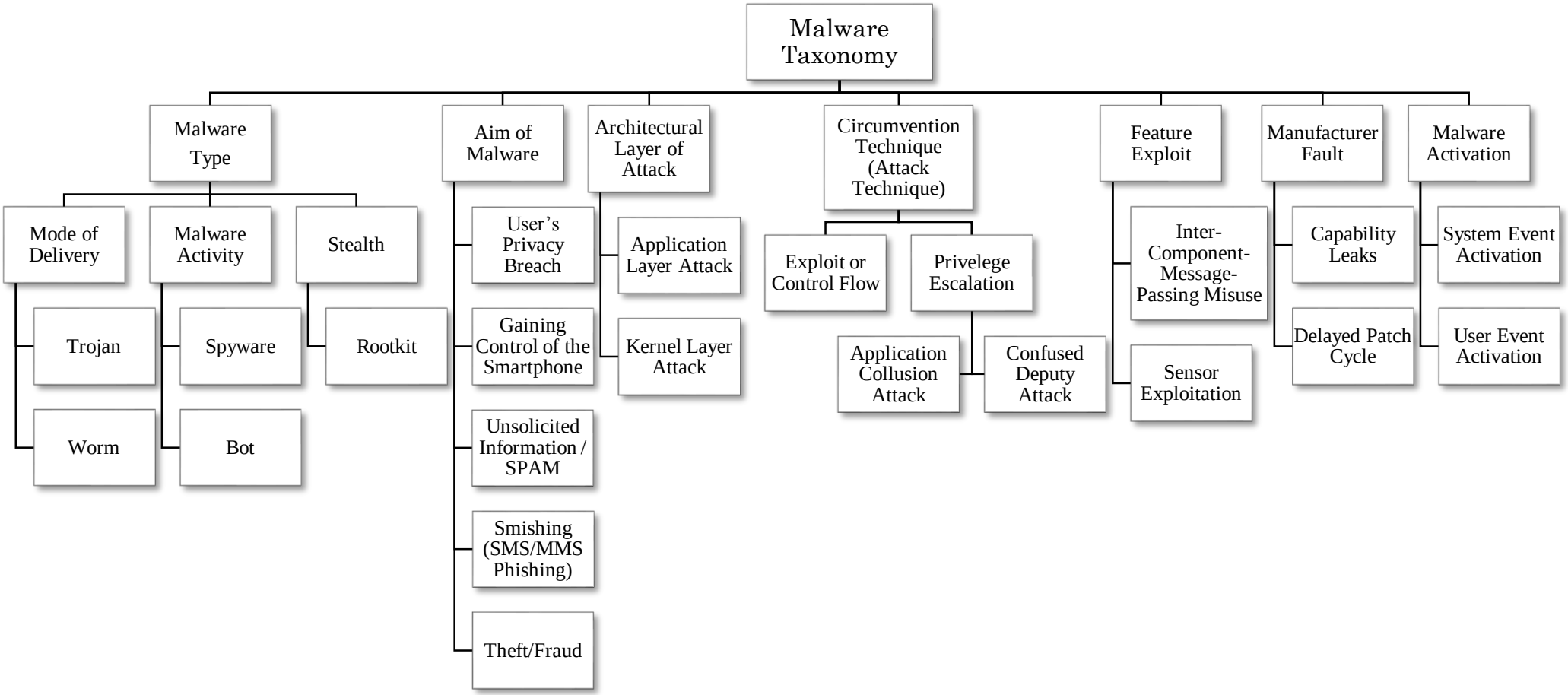

Figure II : Smartphone Malware Taxonomy

Chapter 5

RESULTS AND DISCUSSION

The true function of philosophy is to educate us in the principles of reasoning and not to put an end to further reasoning by the introduction of fixed conclusions.

– George Henry Lewes

5. RESULTS AND DISCUSSION

5.1. Smartphone Malware Lifecycle

Figure III depicts that malware developed for smartphone devices undergo distinct stages during their whole span of generation and existence. In order to be able to limit the damage or prevent the attack at any particular phase, it is cardinal to comprehend all these stages. Initiation of malware starts as a result of a developer's or a group of developers' illegitimate intentions to attain illegal/illegitimate objectives. This results in the process of either developing a new instance of malware or modifying any current instance sample of malware (mostly known as repackaging). For example, given the situation of Android, launching a malware application on Google Play Store was trivial before commence of Google Bouncer. However, malware developers are utilizing various social engineering techniques to persuade users to spread their malware using 3rd third party application stores. Once a malware is downloaded and installed, it gets triggered on various events to perform its malicious activity. This process of development and propagation repeats over and over again for every new or modified instance of a malware. Each stage is explained further in following sections:

5.1.1. Development or Enhancement

As smartphone sales are increasing along with the usage of their applications; development and repackaging of new malware is also growing at the same pace. Some of main objectives of malware developers include financial gain, fame, theft/fraud, privacy breach and surveillance and gaining control of smartphones etc. Given any objective of malware, development phase is important due to the reason that it has to be done while keeping in mind the current security mechanism so that are required to be circumvented in order to harm user [123] . Another important aspect of development phase is that developers are using cross platform development frameworks like PhoneGap [124] and Xamarinaid [125] to development malware which could run on more than one smartphone platforms, hence one malware may target multiple platforms. Lower costs to get licenses to develop and publish applications on markets are also engages to write smartphone malware.

The development of malware is different from the other applications developed for the smartphone with respect to the pre-attack techniques they need to implement. Unlike benign

application, which provide value added service; the malware applications have to circumvent the defense strategies deployed on the smartphone without which they cannot perform their task. Developing an application with such a perspective and suppositions about the loopholes in the targeted system is risky. Sometimes, the assumptions made cannot prove to be sufficient to overcome the security model.

Even then the developers do not refrain from trying their luck on grabbing any monetary benefits they can and thus the number of malware applications is increasing continuously. The Operating system with the open source models has been the primary target of such malicious software. As stated in the reports published by McAfee and Symantec [88] [126]; out of all the existing smartphone malware, Android has 90% on its plate.

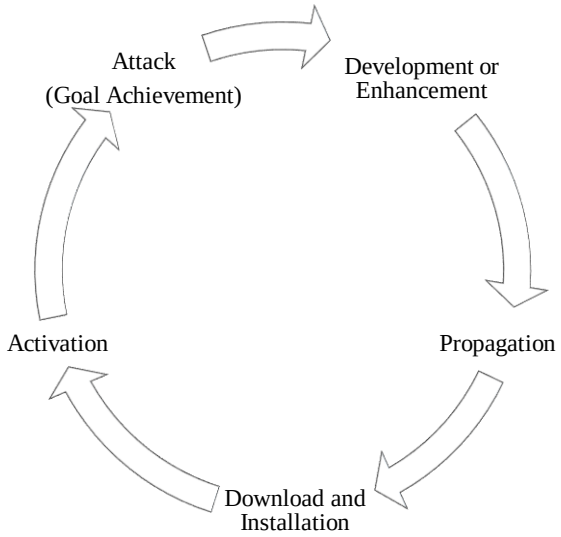

Figure III: Smartphone Malware Life Cycle

5.1.2. Propagation

Once the malware is developed from scratch or enhancements have been deployed on it, the next phase in the plan of action is the propagation. To confuse the user, repackaging the application in a bundle of applications which include one or two useful ones, is the technique which has become very common these days. It is not limited to the smartphones only in fact, software available for use on PC which are available for free download from the internet also come in a bundle. While downloading the application desired by the user, the package installs sniffers or other applications having malicious intents. Propagation stage approaches once a

malware is either developed from scratch or is enhanced from its previous version. There are a number of propagation techniques and among them; social engineering is the most effective and practical. Repackaging, as discussed in the case of smartphones in smartphones in particular have been discussed in [29]. Along with repackaging, some other techniques of propagation can also be used such as drive by download etc., is an effective form of social engineering. It is done by embedding malicious payload into any popular application and republishing it; mostly as a freeware on a third3rd party application store. Malware developers used to publish repackaged applications on Google Play, but precautionary measures taken by Google in recent years have restricted such publications. In addition to repackaging, there are other techniques as well which are being success fully employed to propagate malware; such as drive by download, phishing, etc.

5.1.3. Download and Installation

According to a survey by Zhou, Jiang [29] and Enck *et al.* [127]; the number of malware samples found on official smartphone markets is far less than that of the ones found on 3rd third party markets. As discussed earlier, some popular paid applications are available on 3rd third party application markets for free; which lure users into downloading and installing them. This download and installation process is a commonly adopted approach for the propagation of repackaged malware. Once downloaded and installed, these applications affect smartphone users in their own ways. The smartphone user is deceived into downloading the repackaged malware and in their utter ignorance they install the application in their smartphones which is necessary for malicious activity. Without residing on the smartphone as an application having access to the resources, not accessible otherwise, the malware can never achieve its goal. To avoid such deceit the operating system developer have now introduced a system by which the application asks the user for his consent on using certain critical resources such as internet, phonebook etc. However, it depends on the user to make a sensible and safe decision.

5.1.4. Activation of the Installed Malware (Taking control)

Malware is activated when it is triggered to perform its intended operation. Depending on the type of the application (masqueraded by the malware), activation varies from one malware to another. Malicious payload embedded in an application which runs in the background gets

activated at boot time mostly, whereas, the payload embedded in an interactive application gets activated when the user performs a particular action on that application such as pressing a button, opening a view etc. Zhou and Jiang Xuxian in [29] has a fine-grained division on malware activation discussed a lot of activation techniques used by malware. Activation of a particular malware depends upon its type. Main categories are the ones, which are activated at the boot time, which mostly run in the background. Other type is the one, which is interactive and is activated by any event such as pressing certain button, clicking certain icon etc. No matter what the source of activation is, the phenomena itself is important for a malware to act and perform its desired activity. An un-activated malware is like paralyzed and cannot do anything on its own.

5.1.5. Attack (Goal Achievement)

During this phase, malware launches the attack and compromises smartphone's security features. Malware's usual targets are: user's privacy attributes (e.g. IMEI, GPS etc.), smartphone control (e.g. sending SMSs to premium numbers etc.), downloads of a malicious update, etc. The damage done depends upon the type of malware e.g. if it is a spyware; it sends out the information to a 3rd third party server or its controller, if malware was to make the smartphone a bot in a botnet; it contacts its C&C, etc. This the time when the activity the malware was destined to perform is carried out by them. Depending upon the purpose of the malware the security is breached and actions such as sensitive information leakage through sound device exploitation, touch screen etc. are carried out. Denial of service attack and fake calls which results in the loss of credit to the smartphone user are also some of the type of attacks a malware could perform on the smartphone.

5.2. Difference between Smartphone and PC Malware

Smartphone malware have followed a slightly different evolution pattern from PC Based Malware. The primary purpose of the very first instances of PC viruses was to multiply themselves and infect as many files on a host as possible. They used to spread via various media. Later, they evolved into worms and then botnets, aiming to infect as many hosts as they could via network connection. However, smartphone malware do not primarily aim to spread from one host to another, neither most of them tend to infect high volumes of smartphones based on host to host propagation. It is evident from walled garden approach

adopted by Apple that the case studies from PC Based counterparts were thoroughly accounted for while designing the security systems for their smartphones. The aim of the walled garden approach is to prevent users from installing any malicious payload by scrutinizing every application before publishing into Apple's App Store. This certainly has prevented the mass volume infection of devices, although, there are instances of Trojans; which have resulted in infecting iOS and Android devices in recent years [128]. Although, Google's security for smartphone users approach also follows an inspective approach for Android is not as inspective and secure as Apple's for iOS; yet in the form of security measures such as Google Bouncer [129], permission based model and sandboxing are in place; which tend to provide some protection from high volume propagation of infection. Despite all these measures, Android based smartphone users may end up as a victim of mass to malware infection through installation of malicious applications downloaded from Google Play and 3rd third Party App Markets. Unlike the malware, which were developed for the PC, the aim of the malware whose target was the smartphones did not include replicating themselves and spreading to the other devices via network. The smartphone malware was mainly developed to exploit the sensitive information used during certain transaction to perform financial frauds. This difference between the smartphone and PC malware has been endorsed by the Apple's walled green research which stated that that since the applications are scrutinized at all levels, the applications tending to replicated themselves while exploiting their rights to the network cannot get into the market. However, there were exceptions where Trojans were reported to be installed on certain smartphone devices during application download from the app stores.

Based on the experience with PC based malware; techniques have evolved for smartphone platforms, as discussed in previous paragraph;, which render them secure from attack vectors adopted by PC malware. The existence of difference between the smartphone and PC malware can be explained by the fact that the smartphone manufacturers along with the researchers deployed the security mechanisms in the smartphone to defend them against the attack vectors adapted by the PC malware. The constraints of smartphone platforms have also played a role in reducing the number of attack vectors utilized by smartphone malware as compared to PC Based malware. Also, there isn't a lot of variety found in the smartphone malware, which is due to the restricted numbers of resources, which have the potential to be exploited [130]. However, smartphone malware is not completely different than their PC

counterparts. Alazab *et al.* [131] identified some similarities and differences between PC Based and Smartphone malware based on their propagation, objectives, damages incurred, executable publishing and computational powers. They also identified some of the characteristics such as propagation, stealthiest and phishing capabilities etc. which are similar in both the PC and smartphone malware.

5.3. Malware Threat Rating Based On User Concerns

Smartphone malware found in the wild have distinct motives such as breaching user privacy, taking control of the phone or harming the device. With a rapid increase in the malware infecting smartphones noticed in the first quarter of 2013 [88], it has become the foremost concern of smartphone users. The intensity of this concern is directly proportional to the damage to applications' credibility available on markets; as user has to accept all permissions requested by the application to be installed. It is therefore very important to rate the severity level of the harm that a malware can cause to the user, while keeping in view the threats to the user. In this section, the severity of smartphone malware found in the wild, based on user concerns is discussed. The severity is in terms of the extent of threat the user feels while keeping in mind the loss it can cause if a malware succeeds to achieve its aim. Severity of malware is presented with respect to user concerns in a graphical manner in Figure IV. X-axis plots severity level against the instances of malware families. Y-axis plots the samples of collected malware families in the wild and through various sources.

It is important to discuss the process involved in calculating these rankings, as carried out in the following lines. Felt *et al.* [132] carried out a survey of smartphone users to find out what percentage of total participants get VERY UP-SET for particular risks out of 99 possible risks. Hence they calculated the weightage of each of 99 risks. For example, one of their identified risk is: "Sent premium text messages from your phone" and almost 96% users reported to get VERY UP-SET by that. These 99 risks were categorized into 6 major categories, as shown in Table I.

Table I : Risk Categories

RISK CATEGORY	INVOLVED RISKS
Unsolicited Outbound Communication	Text messages are sent and phone calls are made without smartphone users' knowledge and consent
Data/ Information loss	Deletion of contacts, call history, calendar, applications, passwords, etc.
Accessing Important Data/ Information	Unauthorized access to credit card numbers, PINs passwords, phone identifiers, etc.
Sharing/ Disclosing Data/ Information without User Consent	Disclosure of images, emails, location, list of applications, phone identifiers, messages, etc. to another party
Misuse of Resources	Application installation without consent, changing time, draining battery, rebooting phone, generating notifications/pop-ups, etc.
Spamming Purpose	Spamming on user contact list, insert spam text in user's texts, event invitations and marketing, etc.

In order to assign an appropriate percentage to above categories; average of the percentages of risks lying in each category were taken. For example percentage assigned to the category: "Data/Information Loss" is calculated as average of percentages of risks falling in this category. This percentage (assigned to each category) denotes the VERY UP-SET rate for each category as plotted in Figure V.

This thesis identified those malware families in the collected sample that can pose the threat encompassed by above categories. In order to reflect the severity of malware families, they are rated on the scale from 0 to 6 in Figure IV, where the scale value indicates the number of categories a malware falls in. Malware families lying in multiple categories are considered more harmful than those lying in fewer categories. It is termed as Malware Family's threat rating. The significance of this threat rating graph is to direct research groups and anti-malware firms about developing solutions to target most threatening malware families.

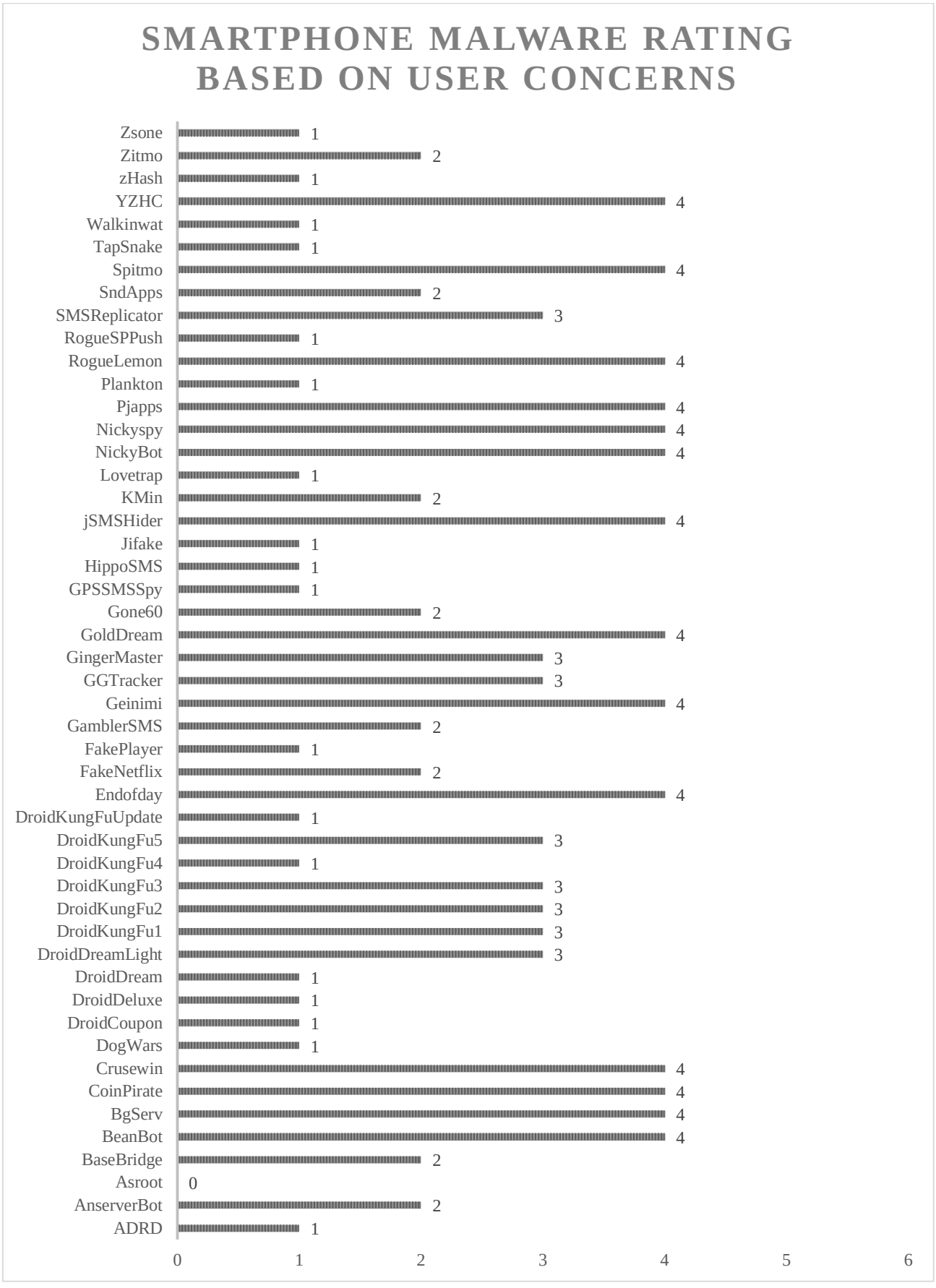

Figure IV: Smartphone Malware Rating based on user concerns

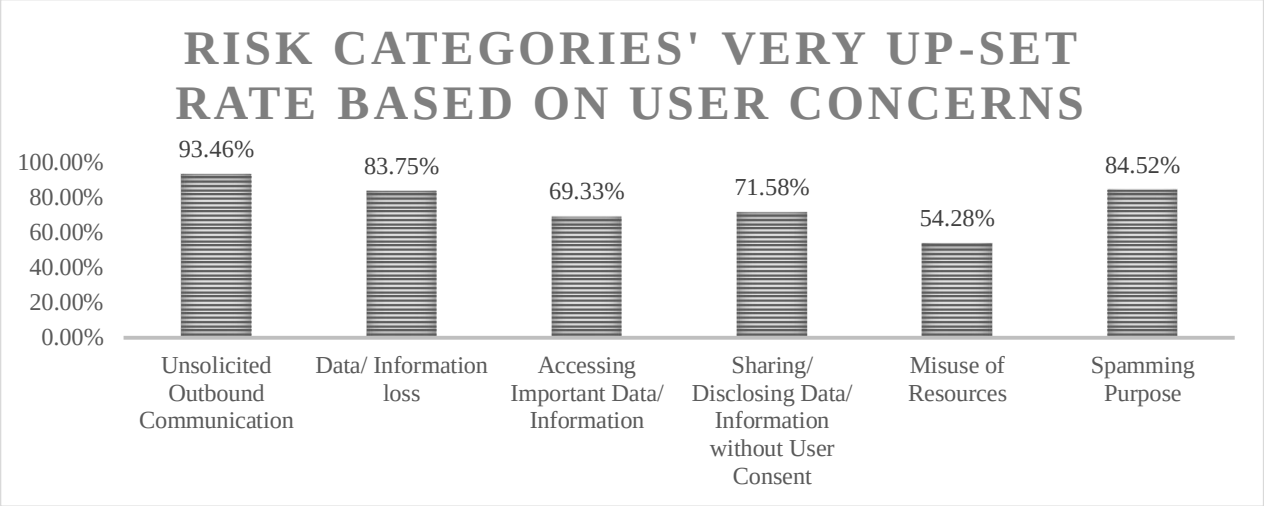

Figure V: Risk Categories' VERY UP-SET Rate Based on User Concerns

5.4. Validation of Malware Dimensions against Malware Instances

This sub-section provides a relationship between various malware and respective malware dimensions from established malware taxonomy. The relationship table (Table II) validates the proposed malware taxonomy by associating instances of malware found in the wild and as proof of concept. It also presents the capabilities of respective malware. Malware occurring in more malware dimensions can be considered as more severe as compared to those appearing against lesser dimensions.

Table II provides a relationship between malware (column headers) and malware dimensions (row headers) from the proposed malware taxonomy. This mapping validates the proposed malware taxonomy by associating instances of malware. Malware occurring in more malware dimensions can be considered as more severe as compared to those against lesser dimensions. Table III provides description of key notations to understand the mappings in Table II.

(sp)iPhone [111] is a sensory malware that reconstructs keystrokes made on a nearby keyboard by exploiting the accelerometer sensor data on the smartphone (cell N10). It could be installed as a result of physical access to smartphone or may be disguised as another application (cell N1). It acts as spyware (cell N3) and provides surveillance about the victim (cell N13) and if misused, can lead to privacy breach that may result in theft or fraud (cell N16). It retrieves sensor data on either regular basis or on system events (cell N11), however

its distance with the keyboard must be minimal (cell N12). GingerMaster [133] enters as a Trojan using repackaging that tricks users into installing it (cell H1). It acts as bot since it communicates with its command and control center to get further instructions and to download root exploit (cell H4). It performs root exploit (cell H8) to gain privilege(s) to get activated and once activated it does various malicious activities which mainly include privacy leakages and gaining smartphone control. It remains undetected as rootkit (cell H5). These activities are performed in response to broadcasts (cell H9) and various system and user generated events (cell H11, H12).

Table II: Malware against Malware Dimensions

MALWARE DIMENSIONS				**MALWARE**														
				Walk & Text	*Fake Player*	*Cabir*	*ikee.A*	*Kakao Talk*	*ikee.B*	*Droid Live*	*GingerMaster*	*Merogo SMS Worm*	*Symbian Sexy Space*	*Soundcomber*	*Contact Archiver*	*Mindtrick*	*(sp)iPhone*	*PlaceRaider*
				A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**
Malware Type	Mode of Delivery	Trojan	**1**	●	●	○	○	●	○	●	●	○	●	●	●	○	◉	○
		Worm	**2**	○	○	●	●	○	○	○	○	●	●	○	○	○	○	○
	Malware Activity	Spyware	**3**	○	○	○	○	●	○	○	○	○	○	○	○	○	●	●
		Bot	**4**	○	○	◉	○	●	●	●	●	○	◉	○	○	○	○	○
	Stealth	Rootkit	**5**	○	○	○	○	○	○	○	●	○	○	○	○	●	○	○
Circumvention Techniques	Privilege Escalation	Confused Deputy	**6**	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
		App Collusion Attack	**7**	○	○	○	○	○	○	○	○	○	○	●	●	○	○	○
	Exploit or Control Flow		**8**	○	○	○	○	●	●	○	●	●	●	○	○	●	○	○
Feature Exploit	Inter-Component-Message-Passing Misuse		**9**	○	○	○	○	○	○	○	●	○	○	●	●	○	○	○
	Sensor Exploitation		**10**	●	○	●	○	○	○	○	○	○	○	●	○	○	●	●
Malware Activation	System Event Activation		**11**	○	○	○	●	●	●	●	◉	○	●	◉	○	●	●	●
	User Event Activation		**12**	●	●	●	○	○	○	○	◉	●	●	●	●	○	◉	◉
Aim of Malware	User's Privacy Breach		**13**	●	○	○	○	●	●	●	●	●	●	●	●	●	●	●
	Gaining Control of the Smartphone		**14**	●	●	○	○	○	●	●	●	●	●	○	○	●	○	●
	Unsolicited Information / SPAM		**15**	●	○	○	○	○	○	○	○	○	◉	○	●	○	○	○
	Theft/Fraud		**16**	●	●	○	○	○	○	●	○	●	●	●	●	●	◉	●
Key: YES ● NO ○ PARTIAL ◉																		

Table III: Taxonomy Mapping Notations

Key	Description
○	Shows absence of a property e.g. in case of defense – malware dimension, it shows the absence of a particular capability of a defense against a malware dimension.
●	Shows presence of a property either clearly asserted or evaluated e.g. in case of interception between defense – malware dimensions, it shows the ability of a particular defense to detect or prevent a malware falling inside a particular malware dimension.
◉	Shows theoretical presence of a property which can be implied from the generalization of attack or defense approach or both e.g. in case of interception of defense – malware dimensions, a partial cell shows the implication that a particular malware family can be detected based on its general behavior since detection of a particular sample from the same family was asserted in the concerned text.

Chapter 6

CONCLUSION

AND

FUTURE DIRECTIONS

Science, history and politics are not suited for discussion except by experts. Others are simply in the position of requiring more information; and, till they have acquired all availaable information, cannot do anything but accept on authority the aopinions of those better qualified.

– Frank Plumpton Ramsey

6. CONCLUSION AND FUTURE DIRECTIONS

6.1. Conclusion

6.1.1. Signature Based Anti-Malware

One of the prominent reasons behind malware spread, especially worm is, that new vulnerabilities are being found and exploited pretty frequently. It takes some amount of time until particular vulnerability is identified, and particular malware's signature is included in signature database. This is apparently due to technique's inherent inability to prevent or detect zero day attacks. Moreover, signature based anti-malware can be deceived by changing the signature of the malware using manual or automated code refactoring.

6.1.2. Smartphone Performance Limitation Issues Aiding Malware

Due to limited resources available on smartphone devices, commercially available anti-malware do not employ effective and state of the art detection and prevention mechanisms. Therefore, one of the key factor behind prevailing smartphone malware are limited resources available on smartphone devices to process and analyze system behavior for effective malware detection. Hence, there is need of extended work and research on the performance efficiency of research contributions toward malware detection in order to mold them towards adoptability.

6.2. Future Directions

6.2.1. Super User Activation

Smartphone operating images normally come with super user account disabled. It implies that privileged tasks can only be performed by system components like kernel and other system services. Applications demanding system level privileges may not run on smartphones. This need to run applications with system level privileges has encouraged smartphone users to activate super user by rooting or jail breaking their smartphone devices. It facilitates users to run privileged applications like file managers, custom themes, etc., while in addition, this makes smartphones more susceptible to malware attacks, especially to rootkits. It can harm users in unexpected ways since smartphones are normally used for social activities, internet banking, and other privacy sensitive purposes. Smartphone vendors, therefore, should highly

discourage super user activation and devise ways to prevent rooting or jail breaking of devices.

6.2.2. User Awareness

As smartphone users belong to different classes of life, likewise varies the awareness level of users. Most users are not aware of consequences of permissions requested by smartphone applications during install time. Moreover, free applications request extended permissions like internet, phone state etc., which are not compulsory for application's functionality. It gets worse in the case of repackaged applications downloaded from 3rd party applications markets. Due to this varying aptitude and proficiency of smartphone users, there is great deal of chance that smartphone user may become victim of social engineering techniques and malware may slip through to the user device. Hence it is important to solve this problem by either creating awareness among users and also by introducing extended security feature of application central certification in order to ensure safety for smartphone users.

REFERENCES

7. REFERENCES

[1] APAC, Canalys, *Smart phones overtake client PCs in 2011*, Shanghai,: www.canalys.com/static/press_release/2012/canalys-press-release-030212-smart-phones-overtake-client-pcs-2011_0.pdf, 2012.

[2] Gartner, Inc., "Gartner Says Smartphone Sales Grew 46.5 Percent in Second Quarter of 2013 and Exceeded Feature Phone Sales for First Time," 14 August 2013. [Online]. Available: www.gartner.com/newsroom/id/2573415. [Accessed 29 August 2013].

[3] McAfee® Labs, "McAfee Labs | McAfee," [Online]. Available: http://www.mcafee.com/us/mcafee-labs.aspx. [Accessed 02 January 2014].

[4] McAfee Labs, "Mobile Malware Growth Continuing in 2013 | McAfee," McAfee, Inc., 21 February 2013. [Online]. Available: http://www.mcafee.com/us/security-awareness/articles/mobile-malware-growth-continuing-2013.aspx. [Accessed 29 August 2013].

[5] Gartner, Inc., "Gartner Says Worldwide Security Market to Grow 8.7 Percent in 2013," 11 June 2013. [Online]. Available: http://www.gartner.com/newsroom/id/2512215. [Accessed 29 August 2013].

[6] Gartner, Inc., "Gartner Says At Least 60 Percent of Information Workers Will Interact With Content Applications via a Mobile Device by 2015," 26 June 2013. [Online]. Available: http://www.gartner.com/newsroom/id/2529315. [Accessed 29 August 2013].

[7] Gartner, Inc., Gartner Identifies Three Security Hurdles to Overcome When Shifting From Enterprise-Owned Devices to BYOD, 04 December 2012. [Online]. Available: http://www.gartner.com/newsroom/id/2263115. [Accessed 29 August 2013].

[8] N. Leavitt, "Malicious code moves to mobile devices," *IEEE Computer*, vol. 33, no. 12, pp. 16-19, 2006.

[9] C. A. Castillo, "Android Malware: Past, Present, and Future," Mobile Security Working Group, McAfee, 2011.

[10] R. Schlegel, K. Zhang, X. Zhou, M. Intwala, A. Kapadia and X. Wang, "Soundcomber: A Stealthy and Context-Aware," in *NDSS*, San Diego, California, 2011.

[11] P. Porras, H. Saidi and V. Yegneswaran, "An Analysis of the ikee.B iPhone Botnet," in *Proceedings of Second International ICST Conference on Security and Privacy in Mobile Information and Communication Systems (MOBISEC)*, Catania, 2010.

[12] C. Funk and D. Maslennikov, "IT Threat Evolution: Q2 2013 - Securelist," Kaspersky Lab ZAO, 2013. [Online]. Available: http://www.securelist.com/en/analysis/204792299/IT_Threat_Evolution_Q2_2013. [Accessed 29 August 2013].

[13] A. Klyubin, "Some SecureRandom Thoughts | Android Developers Blog," Android Developers Blog, 14 August 2013. [Online]. Available: http://android-developers.blogspot.com/2013/08/some-securerandom-thoughts.html. [Accessed 29 August 2013].

[14] Macworld, "Apple unveils iPhone | Macworld," IDG Consumer & SMB, 9 January 2007. [Online]. Available: http://www.macworld.com/article/1054769/iphone.html. [Accessed 29 August 2013].

[15] iPhone (1st generation) - Wikipedia, the free encyclopedia, "iPhone (1st generation) - Wikipedia, the free encyclopedia," Wikimedia Foundation, Inc., [Online]. Available: http://en.wikipedia.org/wiki/IPhone_(1st_generation). [Accessed 29 August 2013].

[16] Wikipedia, the free encyclopedia, "HTC Dream - Wikipedia, the free encyclopedia," Wikimedia Foundation, Inc., [Online]. Available: http://en.wikipedia.org/wiki/HTC_Dream. [Accessed 29 August 2013].

[17] Ars Technica, "Google introduces developer G1 Phones | Ars Technica," Condé Nast, 08 December 2008. [Online]. Available: http://arstechnica.com/uncategorized/2008/12/google-introduces-developer-g1-phones/. [Accessed 29 August 2012].

[18] Brighthand, "RIM Announces BlackBerry 8820 -- Its First Smartphone with Wi-Fi," TechTarget , 17 July 2007. [Online]. Available: http://www.brighthand.com/default.asp?newsID=13186. [Accessed 29 August 2013].

[19] WWW.BBSCNW.COM, "A short history of the BlackBerry - BlackBerry Smartphones," [Online]. Available: http://www.bbscnw.com/a-short-history-of-the-blackberry.php. [Accessed 29 August 2013].

[20] M. Ali, H. Ali and Z. Anwar, "Enhancing Stealthiness & Efficiency of Android Trojans and Defense Possibilities (EnSEAD): Android's Malware Attack, Stealthiness and Defense: An Improvement," in *Proceedings of the IEEE FIT '11, 9th International Conference on Frontiers of Information Technology*,

Islamabad, 2011.

[21] D. H. You, "Android Kernel Rootkit," *Phrack Inc.*, vol. e, no. 68, p. 6, 2011.

[22] Symantec Corporation, "Symantec Security Response | Rootkits," 2012. [Online]. Available: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/rootkits.pdf. [Accessed 12 January 2012].

[23] C. Bower, "Single Process Parasite," *Phrack Inc.*, vol. 14, no. 68, p. 9, 2011.

[24] A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici and S. Dolev, "Google Android: A State-of-the-Art Review of Security Mechanisms," in *arXiv preprint arXiv:0912.5101*, 2009.

[25] M. L. Polla, F. Martinelli and D. Sgandurra, "A Survey on Security for Mobile Devices," in *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, 2012.

[26] ZDNet, "The future of mobile malware - digitally signed by Symbian? | ZDNet," CBS Interactive, 23 July 2009. [Online]. Available: http://www.zdnet.com/blog/security/the-future-of-mobile-malware-digitally-signed-by-symbian/3781. [Accessed 2013 29 August].

[27] S. Fahl, M. Harbach, T. Muders, M. Smith, L. Baumgärtner and B. Freisleben, "Why Eve and Mallory Love Android: An Analysis of Android SSL (In)Security," in *In Proceedings of the 2012 ACM conference on Computer and communications security, (pp. 50-61)*, Raleigh, North Carolina, USA., 2012.

[28] InformationWeek.com, "New Smartphone Worm Spreads Via MMS, Bluetooth - InformationWeek," 04 April 2005. [Online]. Available: http://www.informationweek.com/new-smartphone-worm-spreads-via-mms-bluetooth/d/d-id/1031600. [Accessed 26 April 2014].

[29] Y. Zhou and X. Jiang, "Dissecting Android Malware: Characterization and Evolution," in *IEEE Symposium on Security and Privacy*, 2012.

[30] Kindsight | Alcatel-Lucent, "Malware Analysis: Walk&Text Trojan for AndroidOS | Kindsight," 22 September 2011. [Online]. Available: http://www.kindsight.net/en/blog/2011/09/22/malware-analysis-walktext-trojan-for-androidos. [Accessed 27 June 2013].

[31] A. Lelli, "A Smart Worm for a Smartphone – WinCE.PmCryptic.A | Symantec Connect Community," Symantec Corporation, 13 November 2008. [Online]. Available: http://www.symantec.com/connect/blogs/smart-worm-smartphone-wincepmcryptica. [Accessed 26 April 2014].

[32] C. Guo, H. J. Wang and W. Zhu, "Smart-phone attacks and defenses," in *Third Workshop on Hot Topics in Networks - HotNets-III*, San Diego, CA, USA, 2004.

[33] "F-Secure Computer Virus Descriptions: Worm:SymbOS/Mabir.A," F-Secure Corporation, [Online]. Available: http://www.f-secure.com/v-descs/mabir.shtml.

[34] A. P. Felt, M. Finifter, E. Chin, S. Hanna and D. Wagner, "A Survey of Mobile Malware in the Wild," in *Proceedings of the 2011 ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, Chicago, Illinois, USA, 2011.

[35] T. Shields, "Blackberry Mobile Spyware - The Monkey Steals the Berries," VeraCode, Burlington, MA, 2010.

[36] T. Shields, "Blackberry Mobile Spyware - The Monkey Steals the Berries," VeraCode, Burlington, MA, 2010.

[37] Flexispy, Ltd., "FlexiSPY - The worlds most powerful spyphone," [Online]. Available: http://www.flexispy.com/. [Accessed 29 August 2013].

[38] "Mobile Spy | Cell Phone Monitoring Software | Smartphone Spy App," Retina-X Studios, LLC, [Online]. Available: http://www.mobile-spy.com. [Accessed 29 August 2013].

[39] F-Secure Corporation, "Internet Security - Antivirus - Online backup - Mobile Security - Anti-Virus for Mac | F-Secure," [Online]. Available: www.f-secure.com/en/web/home_global/home. [Accessed 30 August 2013].

[40] R. Siilasmaa, K. Alkio and M. Hyppönen, "F-Secure - Malware Wiki," Wikia Community, [Online]. Available: http://malware.wikia.com/wiki/F-Secure. [Accessed 30 August 2013].

[41] Flexispy, Ltd., "F-SECURE is at it again. Review stirs fear mongering and mislabeling about mobile viruses, malware, scams, and other mobile threats.," [Online]. Available: http://www.flexispy.com/fsecure-is-malware.htm. [Accessed 30 August 2013].

[42] Flexispy, Ltd., "FlexiSPY Android Spy GPS location, intercept SMS, Email, Listen, Call Log, Spy Call and Phone Tap Spyphone Software," [Online]. Available: http://www.flexispy.com/remove-fsecure-malware.htm. [Accessed 30 August 2013].

[43] Flexispy, Ltd., "Keep Up With FlexiSPY News," [Online]. Available: http://www.flexispy.com/fsecure-fearmongering-about-mobile-trojans.htm. [Accessed 30 August 2013].

[44] F-Secure Corporation, "Trojan-Spy:SymbOS/Flexispy.A," [Online]. Available: http://www.f-secure.com/v-descs/flexispy_a.shtml. [Accessed 30 August 2013].

[45] F-Secure Corporation, "Spyware:Android/Flexispy.K," 04 September 2012. [Online]. Available: Spyware:Android/Flexispy.K. [Accessed 30 August 2013].

[46] F-Secure Corporation, "Spyware:SymbOS/Flexispy.F," http://www.f-secure.com/sw-desc/spyware_symbos_flexispy_f.shtml, 14 August 2007. [Online]. Available: Spyware:SymbOS/Flexispy.F. [Accessed 30 August 2013].

[47] S. A. Gunasekera, "Analyzing the SS8 Interceptor Application for the BlackBerry Handheld," Zensay Labs.

[48] SS8, Inc., "SS8, Inc. > Products > Overview - Lawful Interception & Communication Forensics," [Online]. Available: http://www.ss8.com/end-end-communications-and-cyber-intelligence-solutions. [Accessed 30 August 2013].

[49] "Etisalat - Home," [Online]. Available: http://www.etisalat.ae. [Accessed 30 August 2013].

[50] G. Bevir, "Etisalat's BlackBerry patch designed for surveillance - Technology - ArabianBusiness.com," Arabian Business Publishing Ltd., 14 July 2009. [Online]. Available: http://www.arabianbusiness.com/etisalat-s-blackberry-patch-designed-for-surveillance-15976.html. [Accessed 30 August 2013].

[51] The Citizen Lab, "Permission to Spy: An Analysis of Android Malware Targeting Tibetans," 18 April 2013. [Online]. Available: https://citizenlab.org/2013/04/permission-to-spy-an-analysis-of-android-malware-targeting-tibetans/. [Accessed 12 April 2014].

[52] R. Templeman, Z. Rahman, D. Crandall and A. Kapadia, "PlaceRaider: Virtual Theft in Physical Spaces with Smartphones," *arXiv preprint arXiv:1209.5982,* 2012.

[53] Damballa, "FirstHalf 2011| Threat Report," Damballa Labs, 2011.

[54] O. Karow, "White Paper: Apple iOS Security in The Enterprise," Symantec Germany GmbH.

[55] A. Lelli, "iPhoneOS.Ikee.B | Symantec," Symantec Corporation, 22 November 2009. [Online]. Available: http://www.symantec.com/security_response/writeup.jsp?docid=2009-112217-4458-99. [Accessed 29 08 2013].

[56] X. Jiang, "DroidLive," NC State University, 2011 November 2011. [Online]. Available: http://www.csc.ncsu.edu/faculty/jiang/DroidLive/. [Accessed 11 April 2014].

[57] C. Papathanasiou and N. J. Percoco, "This is not the droid you're looking for...," in *DEF CON 18*, Las Vegas, 2010.

[58] J. Bickford, R. O'Hare, A. Baliga, V. Ganapathy and L. Iftode, "Rootkits on Smart Phones: Attacks, Implications and Opportunities," in *HotMobile'10*, Annapolis, Maryland, USA, 2010.

[59] NORTH CAROLINA STATE UNIVERSITY, "The Abstract :: North Carolina State University :: Clickjacking Rootkits for Android: the Next Big Threat?," 02 July 2012. [Online]. Available: http://web.ncsu.edu/abstract/technology/wms-jiang-clickjack/. [Accessed 11 April 2014].

[60] X. Jiang, "RootSmart," NC State University, 3 February 2012. [Online]. Available: http://www.csc.ncsu.edu/faculty/jiang/RootSmart/. [Accessed 11 April 2014].

[61] X. Jiang, "GingerMaster," NC State University, 18 August 2011. [Online]. Available: http://www.csc.ncsu.edu/faculty/jiang/GingerMaster/. [Accessed 11 April 2014].

[62] E. Zorrilla, "Root Your Gingerbread Device With Gingerbreak – xda-developers," XDA Developers, 21 April 2011. [Online]. Available: http://www.xda-developers.com/android/root-your-gingerbread-device-with-gingerbreak/. [Accessed 11 April 2014].

[63] Y. Zhou and X. Jiang, "Detecting Passive Content Leaks and Pollution in Android Applications," in *Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, 2013.

[64] E. Smith, "iPhone Applications & Privacy Issues: An Analysis of Application Transmission of iPhone Unique Device Identifiers (UDIDs)," PreSet Kill Limit, 2010.

[65] Research In Motion (RIM), "Protecting the BlackBerry device platform against malware," [Online]. Available: http://www.blackberry.com/solutions/resources/Protecting_the_BlackBerry_device_platform_against_malware.pdf. [Accessed 29 August 2013].

[66] "RIM Company - Learn about Research in Motion," Research In Motion Limited (RIM), [Online]. Available: http://www.rim.com/index_na.shtml. [Accessed 29 08 2013].

[67] "Sites - Global," BlackBerry Ltd., [Online]. Available: http://global.blackberry.com/sites.html. [Accessed 29 August 2013].

[68] F-Secure Ltd., "Trojan:SymbOS/MerogoSMS," [Online]. Available: http://www.f-secure.com/v-descs/trojan_symbos_merogosms.shtml. [Accessed 30 August 2013].

[69] N. Seriot, "iPhone Privacy," December 2009. [Online]. Available: http://seriot.ch/resources/talks_papers/iPhonePrivacy.pdf. [Accessed 29 August 2013].

[70] S. Shekhar, M. Dietz and D. S. Wallach, "AdSplit: Separating smartphone advertising from applications," in *21st USENIX Security Symposium*, Bellevue, WA, 2012.

[71] A. Attaa, "Cellular Companies and Constant SMS Spamming," ProPakistani.Pk, 29 September 2011. [Online]. Available: http://propakistani.pk/2011/09/29/cellular-companies-and-constant-sms-spamming/. [Accessed 30 August 2013].

[72] Wikipedia, "SMS phishing - Wikipedia, the free encyclopedia," Wikimedia Foundation, Inc., [Online]. Available: http://en.wikipedia.org/wiki/SMS_phishing. [Accessed 2013 August 2013].

[73] K. Santos, "Smishing: A Serious Identity Theft Scheme - Yahoo! Finance," Yahoo! - ABC News Network, 25 July 2013. [Online]. Available: http://finance.yahoo.com/news/smishing-serious-identity-theft-scheme-110046141.html. [Accessed 30 August 2013].

[74] T. Espiner, "Phone Trojan 'has botnet features' | ZDNet," ZDNet, CBS Interactive, 16 July 2009. [Online]. Available: www.zdnet.com/phone-trojan-has-botnet-features-3039684313. [Accessed 30 August 2013].

[75] X. Jiang, "Smishing Vulnerability in Multiple Android Platforms," 10 October 2012. [Online]. Available: http://www.csc.ncsu.edu/faculty/jiang/smishing.html. [Accessed 30 August 2013].

[76] M. Grace, Y. Zhou, Z. Wang and X. Jiang, "Systematic Detection of Capability Leaks in Stock Android Smartphones," in *Proceedings of the 19th Annual Symposium on Network and Distributed System Security*, 2012.

[77] M. Dietz, S. Shekhar, Y. Pisetsky, A. Shu and D. S. Wallach, "QUIRE: Lightweight Provenance for Smart Phone Operating Systems," in *20th USENIX Security Symposium*, 2011.

[78] P. Hornyack, S. Han and J. Jung, "“These Aren’t the Droids You’re Looking For” Retrofitting Android to Protect Data from Imperious Applications," in *ACM CCS*, 2011.

[79] AppFence, "AppFence," 2011. [Online]. Available: http://appfence.com/. [Accessed 14 February 2014].

[80] R. Templeman, Z. Rahman, D. Crandall and A. Kapadia, "PlaceRaider: Virtual Theft in Physical Spaces with Smartphones," in *NDSS*, 2013.

[81] A. Mylonas, S. Dritsas, B. Tsoumas and D. Gritzalis, "Smartphone security evaluation - the malware attack case," in *Proceedings of the International Conference on Security and Cryptography (SECRYPT)*, Spain, 2011.

[82] Storm8, Inc., "About Us | Storm8," [Online]. Available: http://www.storm8.com/about-us. [Accessed 30 August 2013].

[83] S. Steve, "Apple distributes Spyware which STEALS your iPhone cell number: Storm8 Games are the applications author with millions of your cell numbers | StopScum," StopScum.com, 27 August 2009. [Online]. Available: http://www.stopscum.com/apple-distributes-spyware-which-steals-your-iphone-cell-number-storm8-games-are-the-applications-author-with-millions-of-your-cell-numbers/. [Accessed 30 August 2013].

[84] McAfee® Labs, "McAfee Threats Report: First Quarter 2012," 2012. [Online]. Available: http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q1-2012.pdf. [Accessed 7 May 2012].

[85] McAfee® Labs, "McAfee Threats Report: Second Quarter 2012," Santa Clara, CA, 2012.

[86] McAfee® Labs, "McAfee Threats Report: Third Quarter 2012," Santa Clara, CA, 2012.

[87] McAfee® Labs, "McAfee Threats Report: Fourth Quarter 2012," Santa Clara, CA, 2012.

[88] McAfee® Labs, "McAfee Threats Report: First Quarter 2013," Santa Clara, CA, 2013.

[89] McAfee® Labs, "McAfee Threats Report: Second Quarter 2013," Santa Clara, CA, 2013.

[90] S. Schrittwieser, P. Fruhwirt, P. Kieseberg, M. Leithner, M. Mulazzani, M. Huber and E. Weippl, "Guess Who's Texting You? Evaluating the Security of Smartphone Messaging Applications," in *NDSS*, 2012.

[91] T. Luo, X. Jin, A. Ananthanarayanan and W. Du, "Touchjacking Attacks on Web in Android, iOS, and Windows Phone?," in *FPS'12 Proceedings of the 5th international conference on Foundations and Practice of Security* , Montreal, 2012.

[92] A. Birsan, "Rootkits On Your Smartphone - InfoSec Institute," InfoSec Institute, 30 July 2013. [Online].

Available: http://resources.infosecinstitute.com/rootkits-on-your-smartphone/. [Accessed 9 April 2014].

[93] RIM BlackBerry, "KB34458-BSRT-2013-006 Vulnerability in BlackBerry Protect impacts BlackBerry Z10 smartphone software," 11 June 2013. [Online]. Available: http://btsc.webapps.blackberry.com/btsc/viewdocument.do?noCount=true&externalId=KB34458&sliceId=1&cmd=&ViewedDocsListHelper=com.kanisa.apps.common.BaseViewedDocsListHelperImpl. [Accessed 30 August 2013].

[94] RIM BlackBerry, "BlackBerry Protect Login - Download BlackBerry Protect and Sign In - UK," [Online]. Available: http://uk.blackberry.com/devices/features/security/protect.html?lpos=gb%3Abb%3Asearch%3ADevices&lid=gb%3Abb%3Asearch%3ADevices%3ABlackBerry-Protect. [Accessed 30 August 2013].

[95] W. Morgan, "Critical vulnerability in BlackBerry 10 OS - The H Security: News and Features," Heise Media UK Ltd., 30 June 2013. [Online]. Available: http://www.h-online.com/security/news/item/Critical-vulnerability-in-BlackBerry-10-OS-1891338.html. [Accessed 30 August 2013].

[96] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer and A.-R. Sadeghi, "XManDroid: A New Android Evolution to Mitigate Privilege Escalation Attacks," Technische Universität Darmstadt, 2011.

[97] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer and Y. Weiss, ""Andromaly": a behavioral malware detection framework for android devices," 2012.

[98] A. P. Felt, H. J. Wang, A. Moshchuk, S. Hanna and E. Chin, "Permission Re-Delegation: Attacks and Defenses," in *20th USENIX Security Symposium*, 2011.

[99] G. Android, "Activity | Android Developers," Android Open Source Project, 23 August 2013. [Online]. Available: http://developer.android.com/reference/android/app/Activity.html. [Accessed 29 August 2013].

[100] R. Rogers, J. Lombardo, Z. Mednieks and B. Meike, Android Application Development, Sebastopol, CA: O'REILLY, 2009.

[101] A. P. Felt, E. Chin, S. Hanna and D. W. Dawn Song, "Android Permissions Demystified," in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011.

[102] L. Davi, A. Dmitrienko, M. Egele, T. Fischer, T. Holz, R. Hund, S. Nurnberger and A.-R. Sadeghi, "MoCFI: A Framework to Mitigate Control-Flow Attacks on Smartphones," in *Network & Distributed System Security (NDSS) Symposium* , San Diego, 2012.

[103] V. Manjunath, "Reverse Engineering Of Malware On Android," SANS™ Institute, Essex, 2011.

[104] B. Lee, L. Lu, T. Wang, T. Kim and W. Lee, "From Zygote to Morula: Fortifying Weakened ASLR on Android," in *IEEE Symposium on Security and Privacy (SnP)*, Oakland, 2014.

[105] B. Albuquerque, "Processing Ordered Broadcasts | Android Developers Blog," Android Developers on Blogspot, 20 January 2011. [Online]. Available: http://android-developers.blogspot.com/2011/01/processing-ordered-broadcasts.html. [Accessed 27 June 2012].

[106] E. Chin, A. P. Felt, K. Greenwood and D. Wagner, "Analyzing Inter-Application Communication in Android," in *In Proceedings of 9th International Conference on Mobile Systems, Applications and Services (MobiSys)*, Washington, DC, 2011.

[107] A. Armando, A. Merlo, M. Migliardi and L. Verderam, "Would You Mind Forking This Process? A Denial of Service Attack on Android (and Some Countermeasures)," in *Proceedings of 27th IFIP TC 11 Information Security and Privacy Conference*, Heraklion, Crete, Greece, 2012.

[108] W. Enck, "Defending Users Against Smartphone Apps: Techniques and Future Directions," in *Proceedings of 7th International Conference on Information Systems Security (ICISS), December, 2011*, Kolkata, India, 2011.

[109] L. Lu, Z. Li, Z. Wu, W. Lee and G. Jiang, "CHEX: statically vetting Android apps for component hijacking vulnerabilities," in *Proceedings of the 2012 ACM conference on Computer and Communications Security*, Raleigh, NC, USA, 2012.

[110] D. Kantola, E. Chin, W. He and D. Wagner, "Reducing Attack Surfaces for Intra-application Communication in Android," in *Proceedings of the 2nd ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, Raleigh, NC, USA, 2012.

[111] P. Marquardt, A. Verma, H. Carter and P. Traynor, "(sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Chicago, IL, USA, 2011.

[112] A. Bose, X. Hu, K. G. Shin and T. Park, "Behavioral Detection of Malware on Mobile Handsets," in *Proceeding of the 6th International ACM Conference on Mobile Systems, Applications and Services*, New York, NY, USA, 2008.

[113] A. D. and A. R., "Keyboard Acoustic Emanations," in *In Proceedings of the IEEE Symposium on Security and Privacy*, 2004.

[114] M. Vuagnoux and S. Pasini, "Compromising Electromagnetic Emanations from Wired and Wireless Keyboards," in *In Proceedings of the USENIX Security Symposium (SECURITY)*, 2009.

[115] Kaspersky Labs, "Viruses move to mobile phones, 2004," [Online]. Available: http://www.kaspersky.com/news?id=149499226.

[116] Symantec Corporation, "Keyloggers Come to Smartphones," [Online]. Available: http://www.mobilesecurity.com/articles/452-keyloggers-come-to-smartphones.

[117] The Hacker News, "Android SwiftKey Keyboard turned into a Keylogger app," [Online]. Available: http://thehackernews.com/2013/03/android-swiftkey-keyboard-turned-into.html.

[118] T. Vidas, D. Votipka and N. Christin, "All Your Droid Are Belong To Us: A Survey of Current Android Attacks," in *WOOT*, 2011.

[119] T. Vidas, D. Votipka and N. Christin, "All Your Droid Are Belong To Us: A Survey of Current Android Attacks," in *WOOT'11 Proceedings of the 5th USENIX conference on Offensive technologies*, SAN Francisco, CA, 2011.

[120] Google, "Android 2.1 Platform | Android Developers," 15 January 2010. [Online]. Available: http://developer.android.com/about/versions/android-2.1.html. [Accessed 19 April 2014].

[121] PhoneArena.com, "Sony Ericsson Xperia X10 to get Android 2.1 upgrade starting today," 19 April 2014. [Online]. Available: http://www.phonearena.com/news/Sony-Ericsson-Xperia-X10-to-get-Android-2.1-upgrade-starting-today_id14286. [Accessed 31 October 2010].

[122] J. Blasco, "Analysis of Trojan-SMS.AndroidOS.FakePlayer.a | AlienVault," AlienVault, Inc., 04 August 2010. [Online]. Available: www.alienvault.com/open-threat-exchange/blog/analysis-of-trojan-smsandroidosfakeplayera. [Accessed 5 September 2013].

[123] Gartner, Inc., "Gartner Says Mobile App Stores Will See Annual Downloads Reach 102 Billion in 2013," 19 September 2013. [Online]. Available: www.gartner.com/newsroom/id/2592315. [Accessed 1 February 2014].

[124] PhoneGap, "PhoneGap | Home," PhoneGap, [Online]. Available: www.phonegap.com. [Accessed 12 04 2014].

[125] Xamarin, "Xamarin - Build mobile apps for iOS, Android, Mac and Windows," [Online]. Available: https://xamarin.com/. [Accessed 12 04 2014].

[126] Symantec Corporation, "Symantec Security Response | Rootkits," 2012. [Online]. Available: www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/rootkits.pdf. [Accessed 12 January 2012].

[127] W. Enck, D. Octeau, P. McDaniel and S. Chaudhuri, "A Study of Android Application Security," in *20th USENIX Security Symposium*, 2011.

[128] D. Maslennikov, "Find and Call: Leak and Spam - Securelist," Kaspersky Lab ZAO, 05 July 2012. [Online]. Available: https://www.securelist.com/en/blog/208193641/Find_and_Call_Leak_and_Spam. [Accessed 19 August 2013].

[129] H. Lockheimer, "Android and Security," Google, 2 February 2012. [Online]. Available: googlemobile.blogspot.com/2012/02/android-and-security.html. [Accessed 20 February 2012].

[130] Lacoon-Security, "Lacoon Mobile Security » Malware Evolution: PC-based vs. Mobile," [Online]. Available: www.lacoon.com/malware-evolution-pc-based-vs-mobile-2.

[131] M. Alazab, A. Alazab and L. Batten, "Smartphone Malware based on synchronization Vulnerabilities," in *The 7th International Conference on Information Technology and Applications (ICITA)*, Sydney, 2011.

[132] A. P. Felt, S. Egelman and D. Wagner, "I've Got 99 Problems, But Vibration Ain't One: A Survey of Smartphone Users' Concerns," 2012.

[133] X. Jiang, "GingerMaster," NC State University, 18 August 2011. [Online]. Available: www.csc.ncsu.edu/faculty/jiang/GingerMaster. [Accessed 11 April 2014].