Mobile Malware Evolution, Detection and Defense

Srikanth Ramu
The Institute for Computing, Information and Cognitive Systems (ICICS),
University of British Columbia
Vancouver, BC V6T 1Z4
Canada
sramu@mss.icics.ubc.ca

Abstract

Use of smartphones has increased exponentially and we are increasingly relying on smartphones for operations like accessing online information, making payment, playing games, using utility applications etc. that were once performed only by computers. The aforementioned operations are besides storing personal details like contact information in Address book, text messages etc. and business data. We are in a new era, where wide range of devices exchange data with each other thus opening up new security concerns. The tremendous growth of smartphone usage makes it a target for malicious attackers to propagate malware and perform other malicious attacks. This survey paper provides an overview of evolution for mobile malware, attack vectors, detection methodologies and defense mechanisms that are still in its infancy stage. This survey paper highlights the unique aspects of mobile malware when compared with PC security and researches that are done to mitigate them. Also, given the popularity of some mobile platforms amongst users, this paper focuses on security mechanisms adopted in iPhone and Android devices to prevent attacks.

Keywords

Smartphones, Mobile, Malware, Android, iPhone, Threats, SMS, MMS, Antivirus (AV), IMEI

1. Introduction

Smartphones adoption is rapidly increasing which is directly linked to the improved computational power and other utility functions. According to Garter [1], Sales of Mobile devices grew 5.6 percent in Third Quarter of 2011 whereas smartphones sales increased 42 percent. Interestingly, Android OS account for more than 50 percent of smartphones sales. Modern day sophisticated mobile phones have three capabilities – communication, computing, and sensing. Although these capabilities provide useful service to the users, they also open up serious security and privacy concerns. This notion is complemented by McAfee's Q3 2011 Threat report that 2011 has been the busiest with respect to malware is concerned in Mobile history [2]. As sales of such smartphones soar worldwide, the stage is set for the massive spread of mobile malware. Mobile malware may perform malicious activities like steal data, send credentials to attackers, send premium SMSs to name a few. Section 4.2.3 gives detailed illustration of mobile threat model. Services like Mobile payment to perform mobile banking, money transfer etc. can draw immense interest to malware authors and attacks on such services could be immensely damaging.

As shown in Figure 1 which is taken from McAfee Lab's Q3 2011 Threat Report [3], the total malware count has increased quarter to quarter. Furthermore, the trend in mobile operating system is more alarming as the number of Android malware is increasing quite rapidly. This is represented in Figure 2 which is taken from McAfee Lab's Q3 2011 Threat Report [3]. Mobile malware has evolved in the last decade and all kinds of malware [52] like worms, Trojan horses, other viruses and spyware have been unleashed against the mobile phones.

In 2011, Damopoulos et al. [4] created an airborne and stealth malware called as iSAM [53] to wirelessly infect and self-propagate to iPhone devices. The goal of the malware is to expose the possible vulnerabilities of modern mobile devices and OS. The iSAM malware besides supporting six malware mechanisms illustrated below connects to an iSAM bot master server and updates its programming logic or obeys commands for a synchronized attack. The iSAM architecture has following malware techniques:

a) *Propagation*: Wirelessly propagates to other iPhone devices
b) *Botnet Update:* To update and control the new version of the malware
c) *Data Collection*: Collects stealthily confidential information
d) *Leak*: Sends stealthily a large number of malicious SMS messages
e) *Availability*: Denial of Application Services in the iPhone
f) *Availability*: Denial of Network Services of the iPhone

Sophisticated malware like iSAM, highlight the challenges ahead in designing highly secure mobile devices and need for continuously evolving malware detection and defense systems.

On a similar note, Android devices too have been targeted with malicious attacks.

Recently, in January 2012, Symantec [5] has identified Android.Counterclank - a Trojan horse for Android devices that steals information. This Trojan is packaged in many applications found in the official Android market. The download figures of all the malicious applications suggest that Android.Counterclank has the highest distribution of any malware identified so far this year.

Zeus In The Mobile (Zitmo) [28] is a classic example of malware to attack Online Banking's Two Factor Authentication system. Zitmo is a heterogeneous Trojan that infects Symbian, BlackBerry, Windows Mobile, and Android devices. Section 4.4.1 and 4.4.2 analyses iPhone and Android security mechanisms respectively.

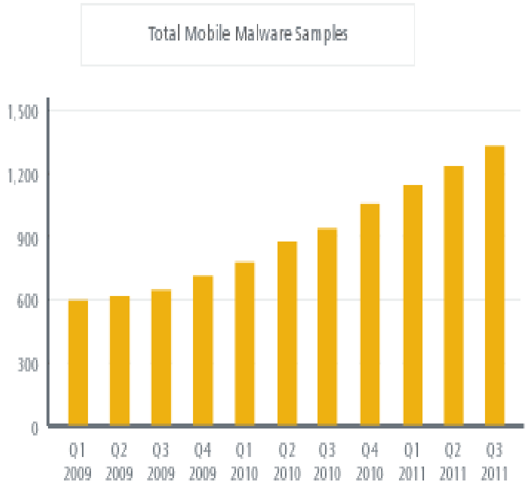

Figure 1- Gartner [1] Report - Quarter wise Total mobile malware samples count

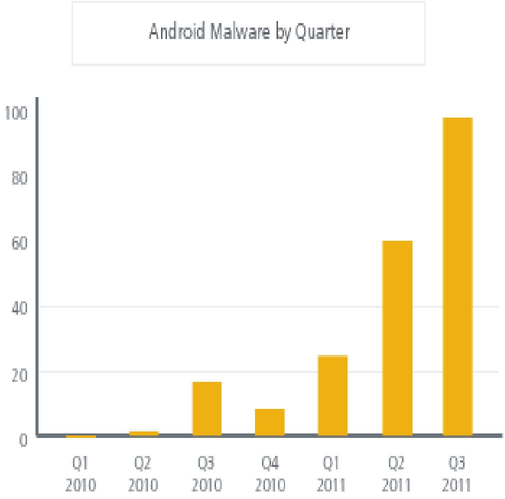

Figure 2- Gartner [1] Report - Quarter wise Android malware count

According to Gostev [35], 2011 witnessed a steep rise in Android malware count. The huge popularity of Android, freely available documentation on Android platform and weak screening process of Android marketplace were attributed to this surge in malware attacks. The report forecasts that there will be surge in malware uploaded to official app stores, especially to Android Market. The report also predicts that Mobile espionage like stealing data from mobile phones, and tracking people through geolocation services will be widespread.

2. Related Work

Mobile malware attacks keeps increasing, more and more researchers are working on studying malware attacks specific to mobile devices. In 2005, Shevchenko [6] presented evolution of mobile malware which is considered to be first comprehensive study. In 2011, Becher et al. [7] continued the evolution from 2005 and explained about specifics of mobile security. The aforementioned study focused on different security classes, however, in this paper we focus primarily on software centric attacks. In 2011, Felt et al. [8] analyzed 46 pieces of iOS, Android, and Symbian malware that spread in the wild from 2009 to 2011. Recently, in 2012 La Polla et al. [9] presented a structured and comprehensive overview of the research on security solutions for mobile devices. Although, initially in our extended abstract paper we did not refer the survey paper from La Polla et al. [9], we later included its study in this paper. This paper carries further research and illustrates latest malwares, detection and defense techniques by referring several papers, blog posts, vendor specifications and tech talks.

3. Initial Definition

Becher et al. [7] define a Mobile phone as a device that can make or receive telephone calls using a smart card controlled by a mobile network operator. Smartphones are mobile devices built with higher mobile computing platform [54] which has an operating system and can have third party applications installed in it. Initially Windows Mobile, Blackberry OS and Symbian operating systems were popular, however, currently iOS and Linux based Android operating systems are instant hit and gained considerable market share. These two operating systems are predicated to dominate the smart phone space for some time. Smartphones permit users to install software applications from sources other than the mobile network operator which requires some controlling to mitigate attacks. In this paper, sometimes smartphones are simply referred as mobile devices or mobile phones. Malware is a malicious code that can do anything in any other program can such as writing a message, stopping a running program, modifying a file etc. Also, malware can be triggered periodically or lie dormant undetected until some event triggers the code to act. They are further classified as Trojans, bots, virus, backdoor, worms, rootkits etc.

4. Discussion

We will start the discussion by briefly summarizing the history of mobile malware in Section 4.1, and then in Section 4.2 we will discuss the specifics of mobile security when compared with computer security and then analyze various attack vector and attack models. We would then take a look at various detection techniques for specific mobile devices in Section 4.3. In Section 4.4 we will analyze the defense mechanisms to control mobile malware. Finally, in Section 5, we forecast the trend in mobile malware space followed by our conclusion.

4.1 History of Mobile Malware

The first malicious software aimed at smartphones hit in 2004. The first virus for mobile phones was written by a group known as 29A in June 2004. An article written by Shevchenko [6], gives a detailed overview of mobile malware history. This first virus was known as '*Caribe*' or Cabir[40] and written for Symbian operating system. Cabir spread via Bluetooth and exploited the limited resource of mobile devices. It shortened the device's battery life by constantly scanning for Bluetooth enabled devices. Subsequently, malwares were written for other operating systems from Windows mobile to the latest Android operating system. All kinds of malwares from file infectors (Virus.WinCE.Duts), backdoor (Backdoor. WinCE. Brador), to Trojans started attacking mobile phones. The propagation of malware was primarily done via Bluetooth, Multimedia Messaging Service (MMS) and Short Message Service (SMS) messaging services. When the article was published, it predicted that the number of malware would increase and rightly so, we are now experiencing this trend. In 2004, Guo et al. [36] described the damage caused by infected smartphones and defense solutions. The paper illustrated the mobile phone specific attacks such as privacy violation, identity theft, emergency call center DDoS, and national crises. This paper is one of the early papers to propose the defense solutions such as hardening approaches, protection at internet and telecom side.

In 2007, Milligan et al. [37] assessed the business risk, threat and countermeasures in using mobile phones. Following are some of the risks illustrated in the report:

- Intentional or unintentional data leakage.
- Data theft
- Business and financial malware attacks
- Network spoofing attacks
- Network congestion by spamming

More recently in 2012, according to Gostev's [35] predication, there would be an increase in Android malware especially financially motivated ones. Thomas et al. [41] discussed the trend of financially motivated malware.

4.2 Mobile Specific Security

Desktop PCs and mobile devices both have similar hardware and software running inside. Hence, security for computers and smartphones has a lot of common characteristics; however, there are some specific aspects that are unique to mobile devices. In 2011, Becher et al. [7] explained the specific characteristics of mobile security. Figure 3 shows the specifics of Mobile security.

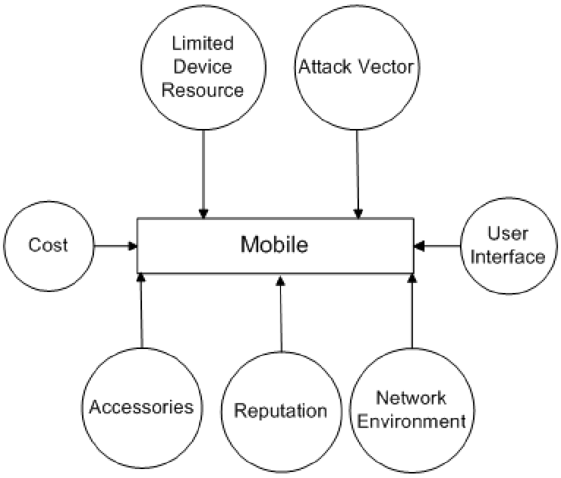

Figure 3 – Mobile specific security

Following are a brief overview of differences of mobile security with computer security:

a) *Limited Device Resources:* Similar to PCs, mobile phones have resources like CPU, RAM, memory, algorithms, battery to name a few. In the past few years, the computational power of smartphones have increased rapidly, however, when compared with computers, smartphones typically have limited device resources. Software applications that run in a computer consuming high resource may not run in a mobile due to constraints in the hardware and software resource. Malware could exploit this and target it by consuming most of the resource and thereby causing denial of service. Moreover, resource constraints also make detection and defense more challenging.

b) *Associated Costs:* One of the motivations for attackers is to make money. In mobile devices, the attacker can generate costs for a user and revenue for the attacker. Attackers use mobile network operator's services like calls, messages, in payment systems like being trustworthy channels as part of the authorization process and incur costs for the user. In 2005, Jamaluddin et al. [10] compared the effects Trojan horses with computers and mobile devices. According to the authors, in PC world Trojan horses impact the speed and performance of the network world, however in mobile world Trojan horses could inflict heavy financial penalty on the consumer. The authors supported their argument by developing a Trojan application that sits inside an application sending SMS or MMS messages, at a cost to the user.

c) *Attack Vector*: Unlike traditional attack vectors related to Desktop PC, mobile devices have some non-traditional vectors which can quickly spread such as SMS, MMS, Bluetooth and traditional IP-based applications.

d) *User interface*: Mobile devices are also different from the desktop PCs in size. Hence, the security mechanisms applied for PCs like visual indicators in browsers, URL bars, CAPTCHAs may not be directly applicable to mobile device. Hence, the may be a need to redesign for smaller screens to suit mobile devices. Also, this calls for greater attention to usable security. Felt et al. [47] illustrate that constraints in mobile user interface makes it easy for attacker to conduct phishing attack than in desktop browsers.

e) *Network Environment*: This is the environment between the device and mobile network operator (MNO). The Network Environment plays a major role in smartphones. Firmware updates process and remote device management and controlled my MNO. This strong influence of MNO over the device brings a new dimension of attack at both the ends. Firmware keeps updating rapidly to keep pace with technology. Due to frequent releases firmware updates are not done locally anymore. It requires MNO to update mobile device with latest firmware. MNOs perform remote management like remote wiping functionality in case the device is stolen.

f) *Reputation*: In case of smartphones, the reputation of MNO plays a key role. When a mobile phone is infected by malware, it might be exploited for malicious activities. However, mobile network operator will charge for every event generated regardless of whether a genuine user action or a malicious trigger. However, from the user's perspective, it is the MNO who charges and not the malicious attacker. This might impact the reputation of the MNO.

g) *Other Capabilities*: Mobile phones are also more vulnerable to unauthorized sniffing on mobile phone sensors. In the case of PCs, sensors are add-on peripherals whereas in present day mobile phones these are part of essential capabilities. In the case of PC, privacy attacks primarily focus on accessing private data and eavesdropping on user activities while interacting with the PC like key loggers. These attacks can be effectively controlled by proper file system access control or encryption. However, in the case of mobile phones, access control on sensors depends on the context, thus making it challenging to defend on privacy attacks. This is discussed in detailed in Section 4.4.4. Recently, Schlegel et al. [42] illustrated a malware that could capture the voice calls and record conversation stealthily in built-in microphones.

4.2.1 Attack Vector and Attack Model

Mobile security threats could be physical or on network connectivity or a malware. *Attack Vector* is a means by which an attacker can gain access to a system. Becher et al. [7] present the attacks to a mobile into following categories:

a) *Hardware Based*: These attacks are more related to physical access of the device such as intercepting mobile network operator smartcard communication. Removing SIM lock of the iPhone and man in the middle attack are some of the examples for hardware centric attack. Attacking the device via debugging functionality is also a type of hardware centric attack.

b) *Device independent attack*: Attacks that are independent of the device such as on infrastructure, protocols etc. come under this category. Global System for Mobile Communications (GSM) protocols were developed 25 years ago and have lot vulnerabilities like immature asymmetric crypto system, no network authentication to name a few. Similarly, there are a lot of flaws in SMS infrastructure like paging channel can overload the network. Flaws in MMS infrastructure causes the batteries to drain quickly.

c) *Software centric*: These attacks are based on exploiting the software running on the mobile devices. As discussed earlier, Cabir malware propagated automatically on Symbian OS in 2004. Some of the software centric attacks using:

- SMS communication channels
- MMS communication channels
- Attacks via mobile web browsers
- Rootkit attacks

d) *User layer*: Attacks that are related to trick the user and not exploiting any technical vulnerability come under this category. Social engineering is a category to lure customers and perform attacks.

The aforementioned paper discusses the counter measures for mobile malware through detection based on signature, static function call analysis, anomaly detection, rootkit detection, and software based attestation. The paper also illustrates protecting the mobile operating system by adopting process of isolation, hardened kernels, secure default settings, software attestation for 3rd party apps.

While analyzing security it is important to focus on the attack model. *Attack model* is analysis of capabilities of an attacker and what are attacker's limits. The attacker can be a passive attacker who does not alter the content or an active attacker who might alter or remove the content. Following are the goals of the attacker:

a) *Eavesdropping*: The attacker gains access to the conversation between the user using the mobile phone and the base station. When an attacker is eavesdropping on a communication, it is referred as sniffing or snooping.

b) *Availability attacks*: The attack which prevents the use of mobile phone by jamming the communication by device and the base station is referred to as Availability attack.

c) *Privacy attacks*: Attacks that focus on getting the information like about location, usage pattern etc. about a user is an attack on his/her privacy.

d) *Impersonation attacks*: It is the ability of an attacker to use the service of MNO without being billed for the usage.

4.2.3 Attacker centric mobile Threat Model

Felt et al. [8] classified mobile threat model into three categories:

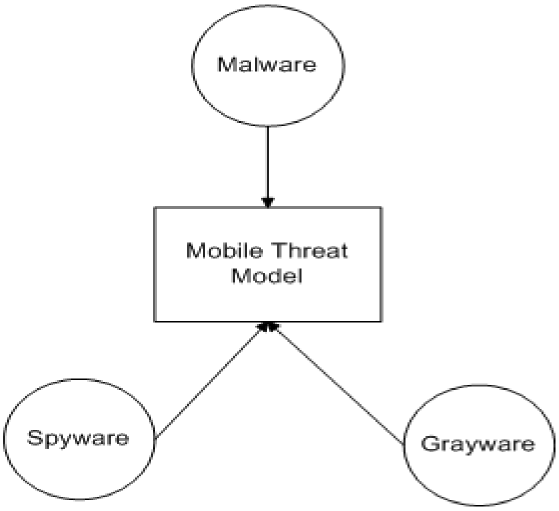

Figure 4 – Mobile threat model

Malware: As discussed earlier, malware gain unauthorized access to the device either by Drive-by download techniques like luring users to install an application or exploiting vulnerabilities in the system like flaws in SMS parser.

Personal Spyware: Personal spyware collects personal information like location, contacts, call history etc. of a user. The attack is carried by gaining physical access to the device and installing the spyware. This attack is more targeted and the data collected is of interest to the person who installed it. Unlike malware, spyware does not send the data to the application developer.

Grayware: Grayware are applications that collect data to be used for marketing and user profiling. The intention behind grayware might not be to harm users. However, sometimes they may behave in a manner that is annoying or undesirable to users.

4.3 Detection

In this section, we analyze various mobile malware detection techniques outlined in various papers. Generally, mobile malware detection techniques can be categorized as *host-based* and *cloud-based*. The technique that runs in mobile phone is termed as host-based technique. However, to improve the efficiency, the intense computation is offloaded to a separate server; this technique is called cloud-based technique. In 2009, Lee et al. [12] compared detection techniques between desktop and mobile devices to highlight energy constraints specific to mobile environment and proposed an energy efficient malware detection technique. Traditionally in PC environment, *Scanning*, *Behavior Checking* and *Integrity Checking* are some of the detection techniques used. *Scanning* is a technique where specific string of bytes are checked against malware format and report vulnerability before it executes in the computer. Unlike Scanning, *Behavior Checking* does not look for malware signature in each file; however, it monitors malicious behavior of an application and detects it. *Integrity Checking* creates log for all the files in the PC along with its details file size, timestamp, checksum etc. The integrity checker runs and examines the files with log and detects of any change. Although these techniques have been widely used for malware detection, each technique has its own pros and cons. In mobile environment, the detection techniques should be energy efficient because of the very nature of limited device resource. Lee et al. [12] proposed such a solution that works under collaboration between mobile and a binary inspection server.

Most mobile-specific versions of antivirus software that is currently available offered by security vendors implement similar techniques used by their desktop variants. Hence, they provide limited detection with significant resource overhead and prove ineffective. On the other hand cloud based detection could do sophisticated threat detection which can be resource intensive. Section 4.3.4 describes cloud based detection system.

4.3.1 Static Analysis:

Analysis of code or application without executing the program is called *Static Analysis*. It is a fast and simple approach. Chandramohan et al. [11] summarized the static analysis techniques suggested in various papers. There are three types of Static analysis which are explained in the below Figures.

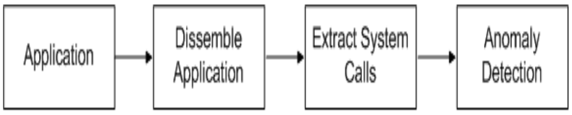

Figure 5(a) - System call based static analysis

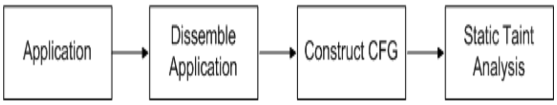

Figure 5(b) - Static taint analysis

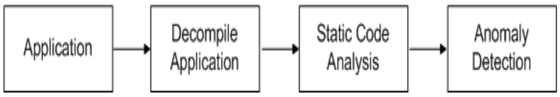

Figure 5(c) - Source code based static analysis

a) System call based
b) Static taint analysis
c) Source code analysis

System call based: The mobile application is first dissembled using tools like IDA Pro. The tools is used to extract the System calls made by the application and then passed to Centroid Machine to perform anomaly detection and classify applications based on the malicious activities.

Static taint analysis: Egele et al. [13] analyzed static taint analysis on iOS application binaries. The study focused on threats posed to users by iOS applications written by third party developers. The study was carried out by developing an automated tool named PiOS that was capable of verifying privacy breaches. The PiOS tool uses *Static Analysis* to check if the application accesses sensitive information and transmit it over the network. Figure 6 illustrates the steps carried in this study. PiOS first creates a control flow graph (CFG) from the application binaries. IDA Pro disassembler is used to extract the binaries and CFG is carried out by:

a) Building a class hierarchy
b) Resolving method calls
    a. Backward Slicing
    b. Tracking Type information

Then reachability analysis is performed on the CFGs to identify the sensitive information that are accessed by the application. To compliment this, a data flow analysis is carried out on the paths from the reachability analysis. Following are some of the data [55] that can be accessed by iOS application:

1) Unique Device ID
2) Address book
3) Current GPS coordinates
4) Photo Gallery
5) Email account details
6) WiFi connection details
7) Call details
8) Safari browser settings and history
9) Keyboard cache

List 1: Sensitive data source

The results of the study show that over half of the applications that were chosen for the study (more than 1,400 iPhone applications) were leaking the Unique ID of the device. The leak of Unique ID helps third party developers to create a finger print of user's application preference and usage patterns. Besides this, there were other sensitive details that were accessed and leaked by some applications. Many applications used common libraries that primarily used to display advertisements to users. Tracking libraries that collect statistics on application users and their usage were also found. The study on the sample set of applications revealed that many applications did leak sensitive data like DeviceID, Location, Address book, Phone number, Safari history and Photos. It is to be noted that from List 1, Email account and WiFi connection details were not accessed. This study also revealed some interesting conclusion about Cydia – a market store for iPhone applications that does not have vetting process and used to install in Jailbreaked iPhones. The study found that applications hosted in Cydia are not more aggressive when compared with App Store.

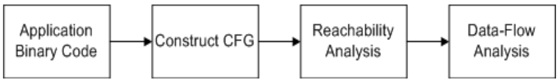

Figure 6 Taint analysis system - PiOS

Source code analysis: Malware detection technique proposed by Enck et al. [14] is illustrated in Figure 5(c). The paper broadly categorizes the security of applications in the Android Market. The study involved implementing a Dalvik decompiler *ded* to reconstruct an application's source code. Then 21 million Lines of Code were analyzed for vulnerabilities focusing on *Control Flow Analysis*, *Data Flow Analysis*, *Structural Analysis* and *Semantic Analysis*.

The analysis results showed that besides leaking device information like Phone number, International Mobile Equipment (IMEI), International Mobile Subscriber Identity (IMSI) and Integrity Circuit Card ID (ICC-ID), over 50% of applications used ad and analytic libraries. Also, it was found that the application developers do not follow proper secure coding guidelines. For example, sensitive details were written in the

Android's centralized logs. In their study they found no evidence of telephony misuse background recording of audio or video, malicious connections.

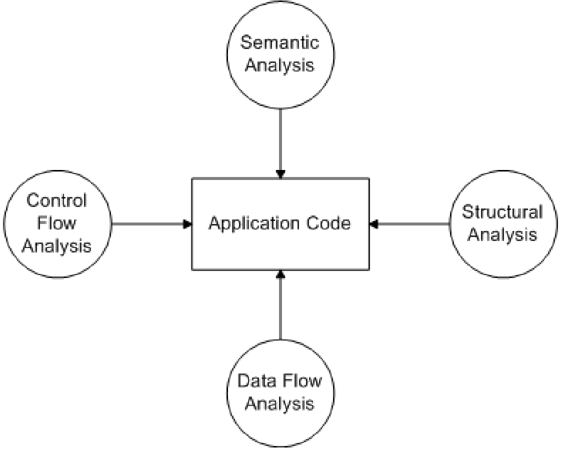

Figure 7 – Source code analysis

Batyuk et al. [15] proposed static analysis detection technique and methods to counter security and privacy attacks. In this on-demand system, the user checks if particular application is malicious one or not, which is available in the Android Market. The requested application is extracted and decompiled. Data mining analysis operations are done by detectors. The user is presented with a report about the detection results. In the experiment the authors found vulnerability related to access, storage and violation of privacy. The decompiled Smali code for some system calls will be similar to all application but differ in register or parameters. In the test conducted by the authors, it was found that many popular free Android applications had privacy violations. They can be mitigated by applying a patch to the decompiled binary without affecting its core functionality. On the flip side, with this type static analysis, it will be difficult to have all the processes running in the phone itself. And static analysis will not be effective when malicious code obfuscates itself.

4.3.2 Dynamic Analysis:

Dynamically monitoring the behavior of mobile application in an isolated environment is termed as *Dynamic* or *Behavioral Analysis*. As the detection mechanisms are improving, malware authors are also getting sophisticated in their attacks. Malware authors test their new malware with existing Antivirus (AV) system so that they remain undetected by AV solutions. Existing Static analysis techniques focus on what is being accessed; besides this Static analysis may yield more *false positives*. However in Dynamic analysis, focus is on why the suspicious operation is performed and how many times it is performed.

In 2011, Isohara et al. [16] proposed a kernel-based behavior analysis for android malware inspection. A brief introduction of Android system architecture is given under Section 4.4.2. The detection system proposed by the authors comprises of a log collector and a log analysis module. The log collector is in the Linux layer and records all system calls. As the log file would increase rapidly, an efficient way of logging important activities is proposed. Process management and File I/O activities are important in the malware context. After logging the activities, the log collector filters events with the target application. In the data analysis module, the log analyzer compares the activities with signatures to detect a malicious activity. The signatures are described by regular expressions. The authors were able to successfully detect malwares using their prototype applications. However, the log analysis module with in the mobile device would still consume huge resource.

Bose et al. [38] proposed a behavioral detection technique to detect mobile worms, viruses and Trojans, as opposed signature-based solutions available during 2008. The study proposed to categorize malware behaviors observing the logical ordering of an application's actions over time. A malicious behavior signatures database is created by studying distinct families of mobile viruses targeting the Symbian OS. A two-stage mapping technique is used to construct the signatures at run-time from the monitored system events and API calls in Symbian OS. The malicious behavior of malware is detected by training a classifier based on Support Vector Machines (SVMs). The study shows that the proposed behavioral detection system could detect malware with more than 96% accuracy.

Ho et al. [39] extended the work done by Bose et al. [38] which was based on runtime comparison between normal and malicious behavior that could be bypassed by obfuscating the behavior. In the study by Ho et al. [39] extend the model by having a filtering system to detect if an event is triggered by a legitimate manual user request or automated request. The entire automated request is screened through a whitelist rules. Furthermore the paper also proposes additional feature to block silent automated transmission attempts.

In 2011, Hsiu-Sen et al. [18] suggested a methodology where various Data Mining concepts were used to detect the behavior of malware. In this study, for behavioral description, ontology is used and for knowledge management - certainty factor theory is used. For automatic detection of mobile malware, fuzzy Petri nets (FPNs) are used.

Ontology is defined as “a formal specification of a shared conceptualization”.

4.3.3 Application Permission Analysis:

Applications run in a sandbox environment however they need permissions to access certain data. At the time of installation, Android platform asks the user to grant or deny permission for the application based on the activities the application can perform. Section 4.4.2 has more description about the permission based security in Android devices. In 2009 Enck et al. [29] proposed Kirin security service for Android platform, to authorize an application to perform sensitive activities. This is to overcome a limitation in Android platform where the developers can intentionally hide permission label to a component. If no label is specified there is no restriction as it had *default allow* policy. The Kirin security service interacts with Android Application installer and it also interacts with collection Kirin Security rules. Rules represent the malicious patterns and it is compared with configuration of the installed application. The study proposes five steps to identify dangerous configurations – (1) Check the phone’s assets, (2) What are the functional requirements, (3) Analyze asset security goals and threats (4) Specify security requirements (5) Analyze security mechanism limitations.

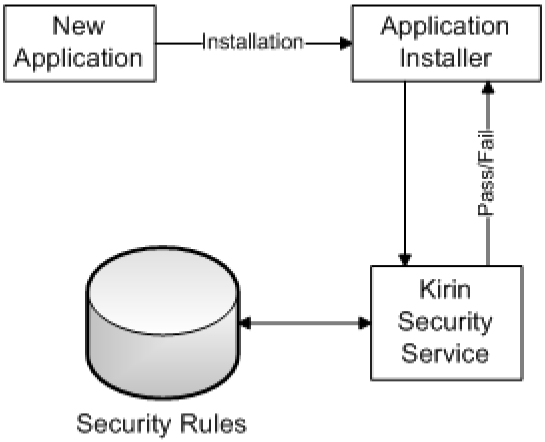

Figure 8 – Kirin Security System

4.3.4 Cloud Based Detection:

As discussed earlier, mobile devices have less resource and having a full-fledged detection system in a mobile device would be a resource overhead. To overcome this, a cloud based approach will be an efficient scheme. In this scheme a light-weight client application monitors the system calls in the device and sends it to the server in cloud to detect malicious behavior. Thus, offloading of powerful computation to the cloud will enable efficient detection for heterogeneous devices. Oberheide et al. [30] argue the advantages of using bandwidth resources and reduce device resources. In the proposed architecture, a *host agent* runs in mobile device that sends the files to a server. Access to each file is captured and the file is checked in a local cache for availability or modification. In case the file is changed or a new file, then it is sent to the server. The second component is the *server* for analyzing the file. The server can have multiple antivirus engines with more sophistication which cannot be done in a mobile phone. The detection could use either Static analysis or Dynamic analysis or both. The server could have an emulator to replay the access to check for any malicious activity. The centralized server could maintain black-listed malware and check for similar pattern in the new files.

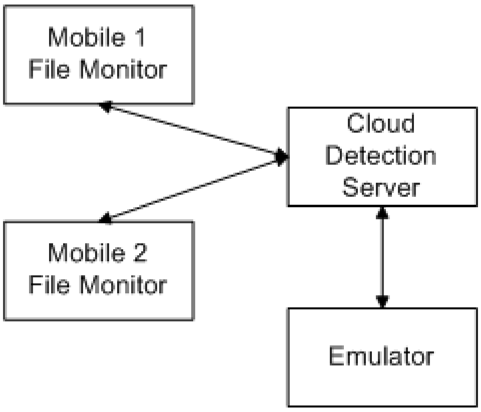

Figure 9 – Cloud Detection System

The advantages of having cloud based detection system are:

a) Efficient detection system through dedicated specialized servers
b) Less usage of device resources
c) Less software complexity at the device

On the flip side, a centralized solutions like cloud based detection needs to be always connected to the cloud to enable live detection. Also, privacy could be a concern as their data is being processed in a central server.

4.3.5 Social collaboration:

In 2011, Yang et al. [17] have illustrated a new malware detection architecture based on social collaboration and used the concept of *hot set*. The study focused on improving existing cloud-based solutions. In the cloud based system discussed in the previous Section, additional hardware for centralized servers and device emulators are required. The hot set concept states that not all malware signatures are equally important. To improve the performance, the hot-set is kept in the phone memory. Each mobile will store the *hot set* signatures for local detection and depend on other social group of mobile device users for *cold sets*.

This approach is termed as Social-AV. The idea is to have a portion of the full signature database in a device and rely on their social group to have a complete signature database. The hot-set in the device is kept up to date with latest signatures and to effectively manage it, it can adopt *Least Frequently Used* and *Least Recently Used* replacement techniques. Moreover, the size of the hot-set is made configurable to enable randomness in hot-sets in the entire devices in social group. The study found out that collaboration based approach enhanced the efficiency by 55% when compared with existing Antivirus systems.

4.3.6 Battery life Monitoring:

As malicious application tends to use most of the battery capacity, an interesting methodology was proposed by Liu et al. [31] to observe energy consumption and detect malware. The authors proposed VirusMeter that detects anomalous behavior by abnormal power consumption. The idea behind this approach is any malicious activity would consume more battery. A user centric *power model* is constructed by recording and characterizes the power consumption of every legitimate activity. VirusMeter monitors the activities in the phone and uses APIs provided by the mobile platform to collect the remaining battery capacity. Based on the collected data it computes how much the application can consume battery and compares it with the power model. If there a difference in exceeds the threshold then it raises alarm. The experimental test results shows that VirusMeter could detect malicious activities with average detection rate close to 80% for various cases. Although, this is a good approach but there are challenges in constructing the power model and collecting real time power consumption. Since, the VirusMeter runs within the mobile phone, it has to be lightweight.

Similar study was carried out by Kim et al. [45] and proposed a malware-detection framework which has good knowledge about power requirements of an activity. The idea is to monitor and analyze with unknown energy-depletion threats. A data analyzer generates a signature for the power usage from the generated history and compares it with detected malicious activity.

4.3.7 Hybrid solution - Comparative Study:

In 2010, Shabtai [19] presented a comparative study of various detection techniques in Android device and conclude that for a comprehensive protection, a combination of various techniques operating in synergic fashion is essential. For example, the authors found that Knowledge-based Temporal Abstraction (KBTA) method has about 94% detection rate with CPU consumption of 3% on average. While the Intrusion Detection Framework used by them had 84% accuracy and 0.126 false positive rate. Static analysis techniques using machine learning classification techniques proved 91.8% accuracy with a 0.172 false positive rate.

4.4 Defense

In this section we present various defense techniques to mitigate mobile malware. To safeguard users and corporate, it is essential to have a defense strategy. The prevention-based system should complement the detection-based system. In the following Sections, we have illustrated various prevention techniques proposed by various researches.

4.4.1 Controlling Malware in iOS:

Miller [20] et al. published a paper on attacks and defenses of iOS and Android devices. One way to control the malware propagation is by offering public market place [49] complimented with an approval process before hosting the application. This is called vetting process and it should ensure that all applications conform to Apple’s rules before they can be offered via the App Store. Apple approves an application by code signing with encryption keys. Accessing the applications via App store is the only way for iPhone devices to install applications. This ensures that only, Apple approved applications that follow Apple’s terms of use can be installed in an iPhone. A central marketplace also helps to remove any application if found suspicious after hosting. Apple can also remove the installed apps from devices as well. Secondly, all application runs in a sandbox environment with limited action privileges. All the applications will be running in less privileged rather than root level.

iOS uses data execution prevention (DEP) and address space layout randomization (ASLR) techniques. iOS also makes distinction between code and data. This reduces attacks of feeding a process as data and then executes it. Lastly, iOS installs software only though Apple authorized services. However, software modules are developed to bypass root privileges and overcome any restrictions. This technique is called *Jailbreak* which is explained below.

Root Exploits: Root Exploits also known as *Jailbreak* are used to circumvent phone’s security mechanisms and by which entire iPhone file system open for use. The prime focus of Jailbreak is to bypass SIM-lock and unlock the device from mobile network operator. They are used by malware authors to take control over the phone and by mobile phone owners to customer the phone to their needs. Unlike PCs, mobile devices especially iOS are targeted specifically in SMS message processing and jail-breaking. Any flaw could make it vulnerable for attacks.

In 2010 Bickford et al. [34], illustrated the threat posed by smart phone *rootkits*. Rootkits are malicious software that stealthily exists in certain process or program with privileged access to a system. They have long been a problem for PCs and with smartphones and their operating system characteristics, rootkit pose a serious security threat to smartphones as well. The paper analyses three example rootkits to exhibit that smart phones are equally vulnerable to rootkits as desktop operating systems. However, the unique interfaces that smartphones expose, such as voice, GPS and messaging, provide malware writers with a new attack vector that might be devastating with respect to security and privacy of the end user. In the first example, a remote attacker uses the rootkit attack to stealthily listen into GSM conversations. In the second example, user's privacy is compromised by making the infected smartphone to send a text message with current location. The third example exploits the power intense services offered by GPS and Bluetooth accessories.

4.4.2 Controlling Malware in Android:

Android has seen a phenomenal success since its release. The huge popularity comes with a price of being targeted by malware application developers. Schmidt et al. [46] described the procedure what they considered to create first malware for Android platform using undocumented Android functions. By creating native Linux applications they bypassed the Android permission systems. Android's security features include:

a) Sandboxing
b) Permissions
c) Malware removal

The Android system architecture consists of an embedded Linux system that is customized. This platform interacts with the phone hardware. The middleware and application API runs on top of this customized Linux. All applications use APIs to interface with the phone. The applications are built using Java and they are executed within a Dalvik Virtual Machine running under a unique UNIX [21]. This sandboxing puts virtual walls between applications and an application cannot access data on other parts of the phone.

Similar to Apple, Android too has a public marketplace [50] to host applications; however, unlike Apple, the Android application can be self-signed. Android uses crowd sourcing to rate the applications by users. Based on user complaint application can be removed from marketplace and remove it from the device as well. This is in contrast with Apples signing mechanisms. The rationale behind Google's self-signing mechanisms it speed up the process of getting the apps developed by the developer in the market quickly. Secondly, Android platform provides a permission-based security enforcement mechanism [22] to protect a resource and data on the device. Access to a system resources and data is controlled during installation time. The permissions required to access the application's resources are defined in its manifest file. During application installation, permission can be accepted or rejected by the user thus delegating the permission management to the user.

In March 2011, Google [23] removed a number of malicious applications from Android market place and suspended associated developer accounts. With the remote application removal feature, Google removed the application from the infected devices and released a security update to protect devices from such attacks. Recently in February 2012, Google [24] released a service codenamed "Bouncer" that scans the applications in the Android market and developer accounts. When a new application is added, immediately the service analyzes for known malware. The Bouncer services checks for the behavior of the application and compares it with known malware. The analysis is done by running the application in a simulated Android setup in Google's cloud infrastructure. Interestingly, this service also analyses new developer accounts to ensure repeat-offending developers are stopped. According to Google, there is 40% decrease in number of potentially malicious downloads from the Android market.

4.4.3 Defense from Proximity Malware:

Defending mobile from proximity malware was presented by Zyba et al. [25]. The mobile phone malware that propagates by proximity contact like direct peer-wise communication mechanisms like Bluetooth or WiFi is termed as proximity malware. Proximity malware might be slower when compared with propagation over the network; however, they may remain undetected by network providers. The authors have presented three techniques to defend mobile devices against proximity malware.

a) *Local detection*: This is simple technique of detecting malware locally and further dissemination is controlled by the device by disabling Bluetooth or WiFi radio. Although, disabling the communication might cause inconvenience to the users, the authors suggest that voice and messaging from the provider would be functional.

b) *Proximity signature dissemination*: Each device maintains a table with signatures of malware files such as MD5 hash of the contents of the file. When the device detects a malware it disables it and propagate it to subsequent devices. The propagation is done when the devices come closer in proximity contact with each other.

c) *Broadcast signature dissemination*: This dissemination technique depends on the mobile network provider to broadcast the signature. Besides unicast messaging, mobile network providers can also broadcast data packets at low cost. In this model, the infected device sends the malicious content to an antivirus server; the server then computes the malware signature and computes a patch to remove the malware from all the infected devices. This technique uses the higher computing power of a dedicated server and also uses expertise of an anti-virus server to compute self "cure" solution.

Zyba et al. [25] illustrated their malware mitigation techniques with their experimental results. They concluded that local detection and mitigation has a marginal impact on propagation. However, a combination of local detection with proximity dissemination of signatures has a dramatic impact on limiting the propagation of malware.

The approach of designing an efficient system to mitigate proximity malware is an ongoing research issue. In this front, Yong Li et al. [26] illustrated a technique to contain malware propagation considering the heterogeneity of mobile devices and resource constraints of the defense system. The study takes two modes of malware propagation namely via MMS and proximity malware via Bluetooth into account. The goals of the defense system are to help the infected node to recover and prevent other nodes from getting infected. Having a centralized patching system requires service provider network to broadcast the signature. The authors have proposed a centralized *greedy algorithm* for the signature distribution problem to be the benchmark and compared it with their *encounter-based distributed algorithm* to disseminate malware signatures. Greedy algorithms are simple approach where decisions are taken based on the information at hand and thinking too much on future consequences. A distributed algorithm approach relies on different actions taken concurrently. The authors choose simulated annealing technique called Metropolis sampler. When two nodes meet each other then there is a configuration change of the algorithm i.e. a node (i) computes a new signature based on the configuration of the encountered new node (j). Then it compares with the signature from its own buffer and chooses to replace its buffer. The replacement of the new signature depends on the acceptance probability. The study analyzes the malware propagation model considering each malware will affect different classes of nodes and number of nodes a malware can infect. This approach focuses on helper nodes that help in propagating the signatures and limit the propagation of malware. The study concludes by comparing the efficiency of greedy algorithm and the proposed encounter based distributed algorithm. By real and synthetic-trace driven simulations, the authors illustrate that their distributed algorithm approaches the optimal system performance.

4.4.4 Defense against Sensor sniffing attacks:

In 2009 Cai et al. [27] proposed a defense system against sensor sniffing attacks where attackers snoop on users by sniffing on mobile phone sensors. As mentioned in the introduction, mobile phones are now having sensing capabilities of audio, video and locations in the form of microphones, cameras and GPS receivers. These additional capabilities open up privacy concerns. The authors developed an attack model to highlight limitations in mobile phones and then propose a framework for preserving privacy of the users. The proposed defense framework consists of three modules:

a) *Policy engine and application monitoring*: The defense system should include effective policy (Whitelisting and blacklisting) and monitoring system.

b) *Interceptor*: The system has an interceptor in between the sensor and the application. When an application violates the access control, the interceptor could take to mitigation actions – Locking and blocking, thus denying the application

c) *User interaction*: Interaction is done by User authorization by asking the user whether his/her has privilege to perform the operation. Secondly, notifying users when a sensor is accessed will make the user aware of it.

The proposed defense mechanisms are:

a) *Context-aware application profiling*: Based on the user's context an application would be given access to the sensor.

b) *Leveraging existing user interaction*: Based on the existing user interactions like picking up the call and ending the call, between these two operations microphone sensor should be accessible.

c) Encryption: Need to encrypt both security and reliable sensory data

4.4.5 Defense based on attacker motivation:

Felt et al. [8] have analyzed defense techniques based on following user motivation.

a) *Selling user information:* Money is one of the main motivations for an attacker. Selling user details to advertising companies is a lucrative option. Mobile platforms need to be hardened to leak information to applications. For example, IMEI theft could be avoided by supporting alternate unique identifier for the devices that are shared to applications. Furthermore, restricting access rights between different applications would improve unauthorized access of data across different applications.

b) *Stealing user credentials*: Stealing user credentials from other applications or SMS could be avoided by isolation mechanism of the applications.

c) *Premium-Rate calls*: User confirmation for a premium rate messages would help user to be aware of the cost.

4.4.6 Data centric security:

Unlike PCs people always carry mobile phones with them and through mobile phones both sensitive and not so sensitive data ranging from personal to business data is being accessed. In 2011, Dehghantanha et Al. [32] proposed a data centric security mechanism to ensure confidentiality, integrity and availability of data stored on mobile devices.

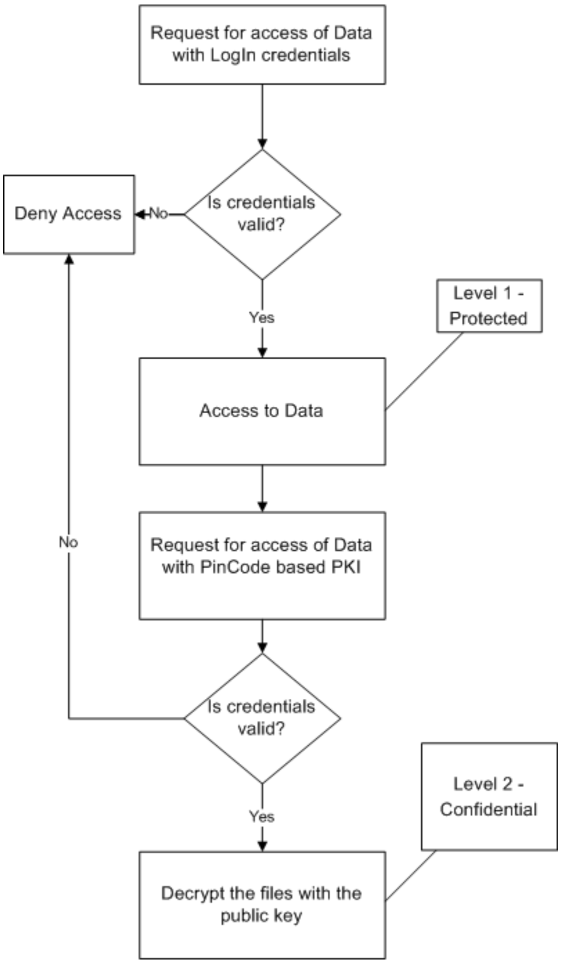
Request for access of Data with LogIn credentials
Is credentials valid?
No
Deny Access
Yes
Level 1 - Protected
Access to Data
No
Request for access of Data with PinCode based PKI
Is credentials valid?
Yes
Level 2 - Confidential
Decrypt the files with the public key

Figure 10 – Data Centric technique

The idea behind this study is to protect the data rather than protecting the device. The study proposes Multi-level data-centric model. The authors argue that categorizing the data like Protected, Confidential, Corporate secret etc. will enable securing the data with various degrees of access rights. The paper proposes classification at lowest level possible even fragments within a file will be easier to secure them. If the data is a too sensitive like corporate secrets then, any access request from mobile can be denied.

4.4.7 Preventive measures:

To control and mitigate malware, it is essential to have complete and comprehensive preventive measures at each level and by each stake holders.

a) *Application Developers:* Application developers need to ensure that they abide by the secure coding [56] and privacy policies. Unnecessary information should not be accessed. For example, instead of using IMEI number, developers can use a unique identifier. Encrypt all the sensitive information that is stored locally or sent to server. For example, using Hash with salt to encrypt the IMEI number. There should be vet mechanism for third-party libraries such as analytics, ad network etc. they use in their application.

b) *Service Level:* At the platform level like application marketplace, proper vetting process should be included to remove suspicious applications. Have a good security policy and incident response plan. Take a zero-tolerance policy.

c) *Smartphone User Level*: Users should ensure that they install a good mobile security solution that can protect and alert for any suspicious events. Download mobile applications from trusted marketplace. Before installing an application, it is essential to research about it by reading their reviews, ratings etc. Pay attention to the permissions requested by the application. Turn off accessory services like Wi-Fi, Bluetooth etc. when not in use. Users should not indulge in “Jailbreak” the system as they are more vulnerable to targeted attacks [48].

d) *Device Level:* At the device level protecting the mobile operating system is required. Security principles like limited privileges and process isolation will restrict violating applications. Hardening the OS by techniques such as Address Space Layout Randomization [51], stack protection, non-executable writable memory etc. Mobile phones should also have sound default settings.

Besides implementing strong counter measures, all stake holders should have a proper response strategy. As Liu et al. [44] showed that how it is possible to perform distributed denial-of-service attacks against critical public services such as 911 using smartphones.

4.5 Mobile Malvertising:

Malicious advertising or use of online advertising to spread malware is termed as *Malvertising*. This technique is used in mobile phones as well. In mobile Malvertising [33], an application would display an Advertisement, when clicked it would redirect to a page where users would be tricked to download the malware. This was done mostly by social engineering. However, Dswani et al. [33] demonstrated Malvertising by drive-by-download. The experiment exploited known vulnerabilities such as Webkit vulnerability, MODE_WORLD_READABLE insecure context etc. When the user visits a compromised web page, the device connects to the attacker via backdoor. Attacker can then issue commands at the device. The demo exploited the Skype version for Android which logged the instant message conversion in clear text.

5. Forecast

Seeing the current trend, it is predicted that the malware count would be increasing with considerable percentage targeting Android platform. According to Gostev [35], attackers would continue writing malware and expanding their focus on variety of exploits. Exploits that are used to escalate the privilege level like rootkits would be widespread. It is expected that 2012 would see its first malware that operates on higher privilege based on drive-by-download attacks. Notwithstanding the vet process and other security measures taken at app store level, more malicious applications would find their way in the marketplace. Emergence of first botnet and mass worm capable of replicating itself for Android platform is expected in 2012. Attackers would next target latest platforms like Windows Mobile 7. Mobile espionage stealing information and targeting specific user will be widespread. The trend is expected to follow the predictions made by Hogben et al. [43] in December 2010 about the attack on data of various classifications such as personal, corporate Intellectual Property, financial assets etc. Risk due to decommissioned devices without removing sensitive data could lead to attackers gaining access to it.

6. Conclusion

Smartphone usage has been rapidly increasing and is increasingly becoming more sophisticated device. The increasing popularity makes them a perfect target for attackers. Smartphones are increasingly being equipped with sophisticated hardware and software systems which open up avenues for sophisticated malware attacks. Smartphones started being targets for malware attack since 2004 and their count is also increasing rapidly. This survey paper starts with describing the evolution of mobile malware with examples of malware for various platforms. We have also outlined threat models and attack vectors for mobile phones. Secondly, we illustrate various detection techniques proposed by various researchers. Finally, we focus on the defense systems proposed to mitigate malware attacks on mobile phones. Although mobile malware classes have some similarity with PC malware, mobile devices have unique characteristics that can be targeted by attackers. Malware attacks cause damage to the users with respect to data theft, privacy, denial of service to name a few. Considering the serious implications malware can cause there should be an effective mechanism to deal with mobile malware. This paper explores the nature of threats to users and organizations.

Just like mobile malware, mitigation techniques have also evolved to catch up with the attacks. In this paper we have discussed both detection-based systems and prevention-based systems. We have highlighted various detection techniques like Static analysis, Dynamic or Behavioral analysis, Cloud based system to name a few. The detection system analyzed covers both signature and anomaly based systems. The control the malware and develop a deterrent system it is essential to understand current security systems adopted by various platform such as Android, iPhone etc. We have analyzed the defense systems in various platforms and also described researches done in the front of defining data centric security systems. Lastly, this paper listed a few trends that are predicted for mobile malware in 2012.

Based on our study on various research papers we propose that all the stake holders have to realize the importance of securing mobile phones from mobile malware. We appreciate various research techniques proposed by various researchers and suggest having a hybrid system incorporating useful aspects of all the techniques discussed in this paper. The intrusion detection system should include thin signature based AV system in the mobile coupled with a server in the cloud to perform extensive detection like behavioral, data mining techniques. Complementing the detection systems, there should efforts to improve prevention mechanisms like hardening the operating system, vetting the application market place etc. Finally, all the users should make themselves educated with the threats and methods to remain safe. It is a reality that mobile malware is widespread and would continue to surge.

7. Acknowledgement

We would like to thank Professor Konstantin (Kosta) Beznosov for reading the extended abstract and providing feedback which formed the basis of extending the abstract into a full survey paper. We would like to thank UBC for providing access to the research papers.

8. References

[1] Gartner Press Release, Egham, UK, November 15, 2011 http://www.gartner.com/it/page.jsp?id=1848514

[2] McAfee Labs Q3 2011 Threats Report Press Release, 2011 http://www.mcafee.com/us/about/news/2011/q4/20111121-01.aspx

[3] McAfee Labls Q3 2011 Threats Report, US, 2011 http://www.mcafee.com/au/resources/reports/rp-quarterly-threat-q3-2011.pdf

[4] Damopoulos, D., Kambourakis, G., and Gritzalis, S. 2011. iSAM: An iPhone Stealth Airborne Malware. In Future Challenges in Security and Privacy for Academia and Industry, J. Camenisch, S. Fischer-H□ubner, Y. Murayama, A. Portmann, and C. Rieder, Eds. IFIP Advances in Information and Communication Technology, vol. 354. Springer Boston, Chapter 2, 17-28

[5] Android.Counterclank Found in Official Android Market http://www.symantec.com/connect/fr/blogs/androidcounterclank-found-official-android-market , 2012

[6] A. Shevchenko, “An Overview of Mobile Device Security” Sep. 2005, http://www.viruslist.com/en/analysis?pubid=170773606

[7] Becher, M.; Freiling, F.C.; Hoffmann, J.; Holz, T.; Uellenbeck, S.; Wolf, C.; , “Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of Mobile Devices,” Security and Privacy (SP), 2011 IEEE Symposium on , vol., no., pp.96-111, 22-25 May 2011

[8] Adrienne Porter Felt , Matthew Finifter , Erika Chin , Steve Hanna , David Wagner, “A survey of mobile malware in the wild”, Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices, October 17-17, 2011, Chicago, Illinois, USA

[9] La Polla, M.; Martinelli, F.; Sgandurra, D.; "A Survey on Security for Mobile Devices" Communications Surveys & Tutorials, IEEE Volume: PP , Issue: 99, 2012 , Page(s): 1 - 26

[10] Jazilah Jamaluddin, Nikoletta Zotou, Reuben Edwards. Member, IEEE, and Paul Coulton, Member, IEEE; "Mobile Phone Vulnerabilities: A New Generation of Malware" 10 January 2005

[11] Chandramohan, M.; Tan, H.;"Detection of Mobile Malware in the Wild", Volume: PP , Issue: 99, IEEE Early Access, 2012

[12] Jong-seok Lee; Tae-Hyung Kim; Jong Kim; "Energy-efficient Run-time Detection of Malware-infected Executables and Dynamic Libraries on Mobile Devices", Future Dependable Distributed Systems, 2009

[13] Egele, M., Kruegel, C., Kirda, E., Vigna, G.: PiOS: Detecting Privacy Leaks in iOS Applications. In: Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS), Feb 2011

[14] Enck, W., Octeau, D., McDaniel, P., Chaudhuri, S.: A Study of Android Application Security. In: Proceedings of the 20th USENIX Security Symposium. August 2011

[15] Batyuk, L.; Herpich, M.; Camtepe, S.A.; Raddatz, K.; Schmidt, A.; Albayrak, S.; "Using Static Analysis for Automatic Assessment and Mitigation of Unwanted and Malicious ActivitiesWithin Android Applications" Malicious and Unwanted Software (MALWARE), 2011 6th International Conference, 2011

[16] Isohara, T.; Takemori, K.; Kubota, A.; "Kernel-based Behavior Analysis for Android Malware Detection", Computational Intelligence and Security (CIS), 2011 Seventh International Conference, 2011 , Page(s): 1011 - 1015

[17] Yang, Liu; Ganapathy, Vinod; Iftode, Liviu; "Enhancing Mobile Malware Detection with Social Collaboration" Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference, 2011

[18] Hsiu-Sen Chiang; Woei-Jiunn Tsaur; "Identifying Smartphone Malware Using Data Mining Technology", Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference, 2011

[19] Shabtai, Asaf; "Malware Detection on Mobile Devices", Mobile Data Management (MDM), 2010 Eleventh International Conference, 2010

[20] Miller, C.; "Mobile Attacks and Defense" Security & Privacy, IEEE, 2011

[21] DalvikVM.com. http://www.dalvikvm.com/

[22] Security and Permissions. http://developer.android.com/guide/topics/security/security.html

[23] Rich Cannings, Android Security Lead, “An Update on Android Market Security” Google Mobile blog, 2011 @ http://googlemobile.blogspot.ca/2011/03/update-on-android-market-security.html

[24] Hiroshi Lockheimer, VP of Engineering, Android, “Android and Security” Google Mobile blog, 2011 @ http://googlemobile.blogspot.ca/2012/02/android-and-security.html

[25] Zyba, G.; Voelker, G.M.; Liljenstam, M.; Mehes, A.; Johansson, P.; "Defending Mobile Phones from Proximity Malware" INFOCOM 2009, IEEE , 2009

[26] Yong Li; Pan Hui; Depeng Jin; Li Su; Lieguang Zeng; "An Optimal Distributed Malware Defense System for Mobile Networks with Heterogeneous Devices"; Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference, 2011 , Page(s): 314 - 32

[27] Liang Cai , Sridhar Machiraju , Hao Chen, Defending against sensor-sniffing attacks on mobile phones, Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds, August 17-17, 2009, Barcelona, Spain

[28] Axelle Apvrille, Senior antivirus analyst and researcher, “Zitmo hits Android”, July, 2011 @ http://blog.fortinet.com/zitmo-hits-android/

[29] William Enck , Machigar Ongtang , Patrick McDaniel, "On lightweight mobile phone application certification", Proceedings of the 16th ACM conference on Computer and communications security, November 09-13, 2009, Chicago, Illinois, USA

[30] Jon Oberheide , Kaushik Veeraraghavan , Evan Cooke , Jason Flinn , Farnam Jahanian, Virtualized in-cloud security services for mobile devices, Proceedings of the First Workshop on Virtualization in Mobile Computing, June 17-17, 2008, Breckenridge, Colorado

[31] Liu, L. G., Zhang, Y, X., Chen. S. “VirusMeter: Preventing your cellphone from spies” In Proceedings of RAID, volume 5758 of Lecture Notes in Computer Science, 2009.

[32] Dehghantanha, A.; Udzir, N.I.; Mahmod, R. "Towards data centric mobile security", Information Assurance and Security (IAS), 2011 7th International Conference on, 2011 , Page(s): 62 - 67

[33] Neil Daswani, CTO Dasiant Inc. "Malvertising & Mobile Malware Madness (+ How to Cap The Mad Hatters)", http://www.youtube.com/watch?v=Y2B3-5_6Elg, Google Tech Talks, 2011

[34] Jeffrey Bickford , Ryan O'Hare , Arati Baliga , Vinod Ganapathy , Liviu Iftode, Rootkits on smart phones: attacks, implications and opportunities, Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications, February 22-23, 2010, Annapolis, Maryland

[35] Gostev, A. March 2012. Kaspersky Security Bulletin. Malware Evolution 2011 @ http://www.securelist.com/en/analysis/204792217/Kaspersky_Security_Bulletin_Malware_Evolution_2011.

[36] Chuanxiong Guo, Helen J. Wang, and Wenwu Zhu. Smart Phone Attacks and Defenses. Proc. of HotNets III, 2004.

[37] Milligan, P. M. and Hutcheson, D. 2007. Business risks and security assessment for mobile devices. In MCBE'07: Proceedings of the 8th Conference on 8th WSEAS Int. Conference on Mathematics and Computers in Business

and Economics. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 189-193.

[38] Abhijit Bose , Xin Hu , Kang G. Shin , Taejoon Park, Behavioral detection of malware on mobile handsets, Proceeding of the 6th international conference on Mobile systems, applications, and services, June 17-20, 2008, Breckenridge, CO, USA

[39] Ho, Y. L. and Heng, "Mobile and ubiquitous malware" In MoMM '09: Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia. ACM, New York, NY, USA

[40] F-Secure, “Bluetooth-Worm:SymbOS/Cabir,” ”http://www.f-secure.com/v-descs/cabir.shtml”.

[41] R. Thomas and J. Martin, “The Underground Economy: Priceless,” USENIX ;login:, vol. 31, no. 6, Dec 2006.

[42] R. Schlegel et al., “Soundminer: A Stealthy and Context-Aware Sound Trojan for Smartphones,” in Network and Distributed System Security Symposium (NDSS), Feb. 2011.

[43] Hogben, G. and Dekker, M. "Smartphones: Information security risks, opportunities and recommendations for users" European Network and Information Security Agency, Greece. December 2010

[44] Lei Liu, Xinwen Zhang, Guanhua Yan, Songqing Chen. "Exploitation and threat analysis of open mobile devices" In Proceedings of ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS) 2009. pp.20~29

[45] Hahnsang Kim , Joshua Smith , Kang G. Shin, Detecting energy-greedy anomalies and mobile malware variants, Proceeding of the 6th international conference on Mobile systems, applications, and services, June 17-20, 2008, Breckenridge, CO, USA

[46] Schmidt, A.-D.; Schmidt, H.-G.; Batyuk, L.; Clausen, J.H.; Camtepe, S.A.; Albayrak, S.; Yildizli, C.; 2009, "Smartphone Malware Evolution Revisited: Android Next Target?", Malicious and Unwanted Software (MALWARE), 2009 4th International Conference, 2009

[47] A. P. Felt and D. Wagner. “Phishing on Mobile Devices” In W2SP Conference, 2011

[48] Graham Cluley, Senior Technology Consultant at Sophos, 2009, http://nakedsecurity.sophos.com/2009/11/03/hacked-iphones-held-hostage-5-euros/

[49] Apple’s official App Store - http://www.apple.com/iphone/from-the-app-store/

[50] Google’s office application market place - https://play.google.com/store?hl=en

[51] Miller, C. "Mobile Attacks and Defense", Security & Privacy, IEEE, 2011 , Page(s): 68 – 70

[52] M. Hypponen, "Malware Goes Mobile", Scientific American, Vol.295, No.5, pp.70-77, 2006.

[53] iSAM: An iPhone Stealth Airborne Malware, Online Material, http://www.icsd.aegean.gr/postgraduates/ddamop/iSAM/iSAM.pdf

[54] Sharon P. Hall , Eric Anderson, Operating systems for mobile computing, Journal of Computing Sciences in Colleges, v.25 n.2, p.64-71, December 2009

[55] N. Seriot. iPhone Privacy. http://www.blackhat.com/presentations/bh-dc-10/Seriot_Nicolas/BlackHat-DC-2010-Seriot-iPhone%2dPrivacy-slides.pdf

[56] OWASP Mobile Security Project - Android https://www.owasp.org/index.php/OWASP_Mobile_Security_Project_-_Android

8. Appendix

IMEI number – International Mobile Equipment Identify is a unique number to identify the device.

DEP – Data Execution Prevention is a feature of OS to prevent applications from executing code from non-executable memory

SIM – Subscriber Identity Module is an integrated circuit that stores International Mobile Subscriber Identity.