Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 6 (4): 325-341, 2014
ISSN 1815-7432 / DOI: 10.3923/rjit.2014.325.341

Android Malware Detection System Classification

Mohd Zaki Mas'ud, Shahrin Sahib, Mohd Faizal Abdollah, Siti Rahayu Selamat and Robiah Yusof

Faculty of Information Technology and Communication, Univeristi Teknikal Malaysia Melaka, Ayer Keroh, Melaka, Malaysia

Corresponding Author: Mohd Zaki Mas'ud, Faculty of Information Technology and Communication, Univeristi Teknikal Malaysia Melaka, Ayer Keroh, Melaka, Malaysia

ABSTRACT

The proliferation of mobile devices and mobile applications in the market has radically changed the way we handle our daily life activities. The rapid growth of mobile device technology has enabled user to use it for various purposes such as web browsing, ubiquitous services, social networking, MMS and many more. To date, Google's Android Operating System has become the most popular choice of operating system for mobile devices since it is open source and easy to use. Despite the rapid growth of Android-based mobile devices in the market, it's also has becoming an ideal place for malware writers. The increase in mobile application in Android has also ignited the possibility of malicious program that can exploit mobile devices and consequently expose any sensitive transaction made by the user. Hence, there is a need to study the anti-mobile malware and mobile malware detection in order to provide an effective solution to defend the mobile user from any malicious threat. Based on these reason this study addresses the current mobile malware detection issues with the intention to thoroughly survey and study the current available detection system for Android-based mobile devices. From the study, we proposed the taxonomy of mobile malware detection that outline and classify the existing Android-based mobile malware detection system. The benefit and constraint of each classification of Android malware detection system are also discussed.

Key words: Android, android malware, mobile malware, malware detection system

INTRODUCTION

The advancement of mobile devices technology has allowed for the prospect of thousands of different mobile applications that community can use with their mobile devices. These has allowed user to access information and services on their finger tips anywhere, at any time. The services offered by the mobile devices has contributed to the increase of Mobile-cellular telephone subscriptions worldwide, according to the International Telecommunication Union (ITU, 2013), the global penetration of mobile user in 2013 has reached to 6.8 billion user, corresponding to 96% and the subscription of mobile broadband has increased to more than 2 billion subscriptions. The outgrowth of mobile devices has also been supported by the enhancement of the Operating System (OS) Technology supporting the mobile devices.

At present, most of the mobile devices are supported by several mobile devices OS namely iOS from Apple, Blackberry, Symbian, Windows mobile and Android by Google. Among these five

popular mobile OS, Google's Android has been dominating the mobile OS market. Android has surpassed the other OS with 80% market share in the third quarter of 2013 (Van der Meulen and Rivera, 2012).

Its open source nature, credibility, performance and ease of customizing has make most mobile user choose mobile devices that supported the Android OS from the others. The growth of Android OS has also contributed to the progression of Android malware in 2013, 98.05% of the mobile malware found by Kaspersky's Lab (Funk and Garnaeva, 2013) was for the Android platform. The effect of mobile malware is lethal, it can steal credential information from the device, sniffing user activity and location, overbilling user by sending random SMS and MMS to contact, launching denial of services attack from user devices and overloading device resources such as memory, battery and storage (La Polla *et al*., 2013).

In the year 2010 malware in general has costs the consumer in United States USD 2.3 billion and cause 1.3 million personal computer to be replaced. Subsequently, in the same year, it is estimated that malware infection has given USD 118 billion financial impact worldwide. As mobile devices technology is now adapting all the personal computer capability, mobile devices are going to have similar event. According to Juniper Network Mobile Threat Center (Juniper Networks, 2011), the effect of mobile malware including exploiting holes in mobile payment that can provide the attacker an immediate USD 10 million profit. With such issues arises, mobile malware has become and emerging threat in cyber security and some countermeasures need to be taken to overcome this matter. Therefore, it is important to develop and improve mobile devices security equally as the computer security especially in finding a mechanism to protect the system and data resources from any kind of intrusion (Sundaram, 1996).

To address this issue, this study reviewed several current researches on mobile malware detection with the intention to have a better understanding in developing an effective solution to defend mobile devices from mobile malware. Since the first mobile device that supports Android OS was launched in 2008, dozens of researches has been done to mitigate Android's mobile devices from malicious application. Hence this study is investigating current research for the possibility of improving malware detection issue by analysing and classifying the existing mobile malware detection method. Through the study and evaluation of the method, the study proposes the taxonomy to outline the mobile malware detection approach in a systematic organization.

ANDROID MALWARE

Malicious software or called malware is purposely written to exploit the vulnerabilities found in a computer system. Malware developer writes malware code for the different purposes which most of it is used for mischievous activity (Robiah *et al*., 2009). The rapid evolution of it signature and behaviour have made it difficult to stop. Previously, Malware such as Virus, Trojan, Worm and Botnet are synonym to desktop environment and rarely found in mobile device. Nevertheless, as the mobile devices technology evolve to a high-end state and able to support complex OS, it has become the malware next target. With the high proliferation of Android OS in the market, the malware targeting Android or refer as Android malware are also rising.

Since the first discovery of Android malware known as Fake player in 2010; malware for Android has grown to an alarming rate. The malicious activities executed by the malware can varies, for instance Fake player is a trojan that hide behind a legitimate movie player application and can sent SMS messages automatically to premium rate number without the user knowledge (Castillo, 2013). Geimini, Pjapss and Hippo SMS are other example of Android malware that

capture or send SMS without the user knowledge. Some other Android malware are more sinister, it can exploit the Android vulnerabilities and gain an administrator or root access, Droiddream, DroidKungfu and BaseBridge are an example of Android malware that come with privilege escalation binary for exploiting Android vulnerabilities. Zhou and Jiang (2012) have investigated 1260 Android malwares and identify each of the malware malicious behaviour. The study shows Android malware can be categorized based on their malicious payload, which are privilege escalation, remote control by contacting command and control server, financial charges by automatically send SMS to premium number and information theft by stealing information such as device information, phone number, contact list and GPS location information.

Similar studies by Felt *et al.* (2011) and Pieterse and Olivier (2012), also have listed several characteristics and behaviours of malware. By comparing the two studies, it can be concluded that Android malware has the following malicious intention.

Stealing device and user credential information: Most of the Android malware are collecting information regarding International Mobile Equipment Indentity (IMEI) number, International Mobile Subscriber Identity (IMSI) number, GPS Location, Phone number, SDK version and Installed package. Android malware also captured user activities such as SMS send or received and call out or in duration.

Communicate with Command and Control (C and C) server: Android malware have the intention to communicate to a C and C server similar to Botnet. Communication can be made through HTTP or SMS. This connection are made for sending captured information from the device and also for updating the malware package or receiving any other malicious information for instance SMS premium number or any other malicious URL.

Sending premium rate SMS and spam: Android malware has the ability to automatically send SMS without the user knowledge. This can be used as a spam or gathering illegal income by charging user when the SMS is send to the premium number.

Search engine optimization: Android malware can have an automatic script to visit ad, for increasing the number of visitor to a website or for the purpose of generating a charge per click without having actual interest in the target of the ad's link. Even though, it seem like it does not cause any harm, this activity can increase mobile data consumption especially to the user who are subscribing for the mobile broadband it will cause them extra charges.

Updating and download package: Android malware can used the communication made to the C and C server to update the existing package into a more malicious intention or even download a new package and installed it automatically in the user devices.

Draining resources: Android malware application can executed a malicious payload that can utilize all the disk storage or memory (RAM) quotas and hogging the CPU.

The increase of Android malware on the market has been substantially disturbing, every new malware on the market come with new features, ability and new intention. The malicious intentions above might only be valid with the current Android malware but as the advancement of mobile

technologies is growing in a rapid pace, it is possible that the consequences of Android malware will be more devastating to the user. Thus a more effective solution must be developed especially in combating a zero-day Android malware.

ANDROID ARCHITECTURE AND SECURITY FRAMEWORK

Android OS was release commercially in 2008, It was developed by Open Handset alliance led by Google and has been updated several times since the initial release. Purposely design for mobile devices, Android is built on top a customized Linux system. Android is organized into five layers (Armando *et al*., 2012), namely, application layer, application framework, Android runtime, libraries and Linux kernel as depicted in Fig. 1. Every operation in Androids is carried out through interaction among these different layers. Linux kernel has the responsibility interacting with the mobile devices hardware. While the interaction between different applications and OS is handled by Inter-process Communication (IPC) mechanism called Binder (Enck *et al*., 2009a). Every application runs as unprivileged user with a unique UID and executed within a Dalvik Virtual Machine (DVM) that provide a sandbox environment for each of the running application. Application in Android is mostly written in Java and converted into Dalvikbycote before its run in the DVM.

Android has also embedded several measures in helping Android user hardening the security of their Android devices. Every application is provided with a manifest file which consist essential information on the application, these include the name of java package, the component of the application and the list of permission acquired by the application. The permission requested give the opportunity to the user to review the permissions the application must have in order to access protected parts of the API and interact with other applications. This can be an indicator whether the application is intended to do any malicious activity or not. However, Felt *et al*. (2012) has

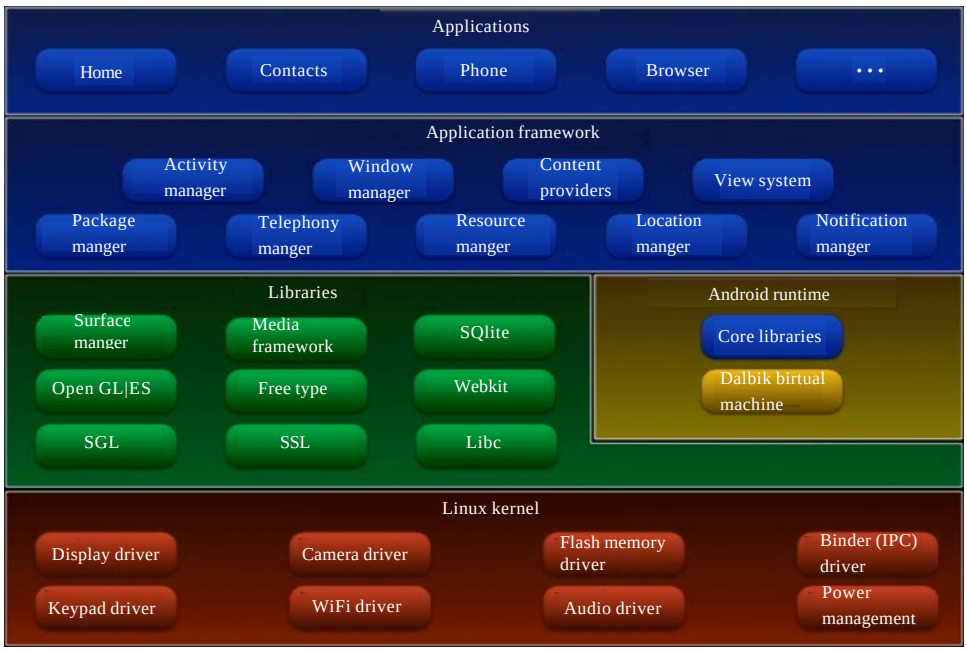

Fig. 1: Architecture of android platform (Armando *et al*., 2012)

indicated that Android permission has failed to inform the majority of Android user due to their lack of awareness, understanding and comprehension towards the requested permission.

In addition, Google's Play Stores also enforcing security measures by scanning its application in the store and blacklisting the application that is found to be embedded with malicious content is a one of a good effort in countermeasures the threat of Malware. Yet, it only covers the application offered by Google's Play Stores, with user are able to download any application from any free application market, this countermeasures seem effortless to combat the Android-based Malware. To overcome this issue, the latest Android version 4.2, codename Jelly Bean, has introduced a feature to verify third party application in its new OS, however research done by Jiang (2012) shows the system has a malware detection rate of only 15.32%.

ANDROID MALWARE DETECTION

Regardless of the physical features different between mobile devices and desktop environment, both devices still access the similar data, application and services. Therefore, similar fundamental concept of defence mechanism can be applied to both of the devices. Yet, one has to consider that mobile devices have a limited storage, memory, CPU and power consumption compare to the desktop environment. Thereby, applying any defensive mechanism to mobile devices has to take these limitations into consideration. Study by Shabtai *et al.* (2010) has listed five threats cluster in Android and it security mechanisms to countermeasures the threats, the five threats clusters are:

- Maliciously using the permissions granted to an installed application
- Exploiting vulnerability in the Linux kernel or system libraries
- Exposing private content
- Draining resources
- Compromising the internal/protected network

Among the defence mechanisms introduce in Shabtai works is Intrusion Detection System (IDS), it works as a tool to detect abnormality or malicious behaviour in a system. Successively, malware detection is considered as a compliment of IDS as a component that identifies malicious activity (Preda *et al.*, 2007). Despite the successfulness of deploying IDSin the desktop environment, adapting IDS in mobile environment requires a well plan design in order to overcome the limitation of the mobile devices. Shaerpour *et al.* (2013) has reviewed several trends of Android malware detection and list the common challenges faced in Android malware detection research. Among the challenges are:

- Polymorphic nature of Android malware that keep changing the behaviour of malware Android
- Issue of false negatives and false positive in detecting Android malware
- Concern of collecting large amount of data in dynamic analysis which exhausting the limited storage resources of a mobile devices

To overcome all the challenges and clearly identify the Android malware detection in depth, it is required to have a more systematic and comprehensive review on existing Android malware detection.

ANDROID MALWARE DETECTION SYSTEM

For the purpose of reviewing the existing Android malware detection research, this study haschoose to classify Android malware detection system into four categories, namely, detection techniques, detection analysis, detection platform and detectionaudit data source.

Detection techniques: Commonly, Android malware detection techniques can be classify into 3 detection techniques that are signature-based detection (SB), anomaly-based detection (AB) and specification-based (SPB) detection (Okazaki *et al*., 2002; Ko *et al*., 1994; Kumar *et al*., 2005; Liao *et al*., 2013). The SB detection detects malware by comparing the application signature or pattern captured with database of known attack or threat. AB detection monitors regular activities in the devices and look for any behaviour that deviate from the normal pattern. Similar to AB detection SPB detection also monitor for any deviation but rather than detecting the occurrence of specific attack patterns, it monitors for deviation of their behaviour from the normal specification. In addition, Pieterse and Olivier, (2012) has also discussed on the fourth classification known as hybrid techniques that either integrating signature with anomaly based detection (SAB) or integrating anomaly with specification based detection (SPAB).

Detection analysis: In obtaining the behaviour of the mobile malware, it is important to understand how the current malware infection will affect the mobile devices. Commonly it is done by analysing the malware sample itself. Farmer and Venema (2004) described malware analysis as a practice to study and to obtain information about the behaviour of a malware in its environment. Subsequently, the result of the analysis will be used as an input to the malware detection. Malware analysis or also known as reverse engineering of malware consists of two approaches, namely, static analysisanddynamic analysis (Distler, 2013). In static analysis, detection is done through the source code, binary or the API level without the execution of Android malware. On the other hand, dynamic detection, detects Android malware through the execution behaviour of the mobile malware. In dynamic detection, the detection is done through monitoring the execution of Android malware activity at runtime.

Detection deployment platform: Malware detection can be deployed either in the host or on a remote server. In host detection, all the activity of the device is monitor, analyse and process in the device itself. Meanwhile deploying detection on a remote server monitors the activity of the device in the device but the analysis and detection process is done on the remote server.

Detection audit data source: Another important categoryin classifying Android malware detection system is the audit data source monitored in the detection process. Among the data source collected in the Android malware detection can be trace within the five Android framework layers, namely, application layer, application framework, Android runtime, libraries and Linux kernel. In addition, network traffic data can also be monitor for any malicious communication activity through the network. These data source usually collected in the dynamic analysis, in which the data is collected during the Android malware runtime. The example of these data source are sensitive data flow, application runtime, system call to kernel, Hardware Performance (CPU, memory or power usage) and network traffic. On contrary, the data source collected in static analysis approach is mostly form the application package itself. The types of data source collected aremanifest file, API and byte-code.

These four categories are used as a parameterin getting a depth understanding on the Android malware detection system. This study reviewed 26 studies related to Android malware detection and based on these categories, this study will proposed Android malware detection system taxonomy to be used as a general guideline in developing an improved Android malware detection system.

DISCUSSION AND ANALYSIS

The 26 Android malware detection researches are compared and categorized based on the classification. Table 1 indicates the classification and comparison between the current Android malware detection study.

Most existing research on Android malware detection system has focused on applying anomaly-based detection technique. From the 26 studies 16 (Arp *et al*., 2014; Abela *et al*., 2013; Dini *et al*., 2012; Atzeni *et al*., 2013; Dixon and Mishra, 2013; Sanz *et al*., 2013a; Moonsamy *et al*., 2013; Gascon *et al*., 2013; Sanz *et al*., 2013b; Aafer *et al*., 2013; Shabtai *et al*., 2012; Wei *et al*., 2012; Wu *et al*., 2012; Burguera *et al*., 2011; Portokalidis *et al*., 2009, 2010) were detecting malware based on the abnormality found in the Android application characteristic gathered from either the dynamic or static analysis. Consequently, eight (Faruki *et al*., 2013; Xu *et al*., 2012; Grace *et al*., 2012; Chan *et al*., 2012; Isohara *et al*., 2011; Enck *et al*., 2010; Enck *et al*., 2009b; Fuchs *et al*., 2009) of the researches were using signature-based detection technique and only one (Dini *et al*., 2013) research was using the specification-based technique. However, Zhou *et al*. (2012) have applied both of the signature-based and anomaly-based detection technique in their Android malware detection system.

The choices made in choosing the detection technique can determined the reliability and effectiveness of the Android malware detection system. For instance SB detection make it possible to detect known attack accurately and using less computational resources but it is less effective to unknown or new malware. Moreover, it is hard to keep the signature up to date and constant update can consume the limited storage the mobile device has. Meanwhile AB detection can effectively detect unknown malware but required more computational resources and produce some amount of false alert. The SPB detection is similar to AB detection, where it detects activity that is deviating from the normal behaviour but instead of relying on statistical or machine learning method, SPB manually developed the specification of the intended behaviour of the system it's monitoring. Deriving the detail specification for SPB is time consuming and there is a chance the detection will produce false negatives due to some malicious activity may be missed. Consequently, Hybrid-based detection is introduced to compensate the advantages and disadvantages of each of the detection techniques.

From the Android malware analysis perspective, 12 of the researches (Faruki *et al*., 2013; Grace *et al*., 2 012; Chan *et al*., 2012; Enck *et al*., 2010; Fuchs *et al*., 2009; Arp *et al*., 2014; Sanz *et al*., 2013b; Moonsamy *et al*., 2013; Gascon *et al*., 2013; Sanz *et al*., 2013a; Aafer *et al*., 2013; Wu *et al*., 2012) were detecting Android malware application with static analysis approach. By using this approach the malicious application can be quickly detected and able toprevent the malicious application from being installed in the device. Yet the static analysis has a drawback in which the approach can be evaded through obfuscation or encryption technique. In contrast, the obfuscation and encryption technique is not able to avoid the dynamic analysis approach as it will look at the behaviour of the malware during runtime. From the survey 13 of the studies (Xu *et al*., 2012; Isohara *et al*., 2011; Enck *et al*., 2009b; Abela *et al*., 2013; Dini *et al*., 2012;

Table 1: Classification and comparison of various android malware detectionsystem

Author	Description	Proposed malware detection system	*Technique	**Approach	***Platform	Data type
Faruki *et al*. (2013)	Detects Android malware by comparing the application with signature generated by extracting statistically improbable features	Andro similar	S	S	H	App. Package
Xu *et al*. (2012)	Repackage Android application in a package that consists of secure policy that intercepts any API call that violates the policy	Aurasium	S	D	H	Kernel
Grace *et al*. (2012)	An automated system called that scalably analyse whether a particular app exhibits dangerous behaviour	Riskranker	S	S	S	App. Package
Chan *et al*. (2012)	An Android application analysing tool which searches for the vulnerability in Android applications by using interprocedural control flow graph searching and static taint checking to detect exploitable data paths in an Android application	Droid checker	S	S	H/S	App. Package
Isohara *et al*. (2011)	Monitor the behaviour of the system call package and compare it with a signature database that is constructed based on Information leaking application detection, Jailbreaking application detection and Destructive application detection		S	D	S	Kernel and Data
Enck *et al*. (2010)	Performs lightweight certification of applications to mitigate malware at install time. The certification rules are constructed purely from security configuration available in application package manifests	KIRIN	S	S	H	App. Package
Enck *et al*. (2009b)	Automatically label data from privacy-sensitive sources and transitively applies labels as sensitive data.	Taint droid	S	D	H/S	Data
Fuchs *et al*. (2009)	Automatic reasoning tool for the security of Android applications that extracts security specifications from manifests that accompany such applications, and checks whether data flows through those applications are consistent with those specification	SC android	S	S	H	App. Package
Arp *et al*. (2014)	Detecting malware by performing a broad static analysis by gathering as many features of an application possible and embedded the features in a joint vector space for automatically find the malicious activity pattern.	DREBIN	A	S	H	App. Package
Abela *et al*. (2013)	Determines malicious behavior from benign behavior through the system call using machine learning techniques	AMDA	A	D	H	Kernel
Dini *et al*. (2012)	Multi-level anomaly detector for android malware that concurrently monitors Android at the kernel-level and user-level to detect real malware infections using machine learning techniques	MADAM	A	D	H	All except App. package

Table 1: Continue

Author	Description	Proposed malware detection system	*Technique	**Approach	***Platform	Data type
Atzeni *et al*. (2013)	Malware detection Framework through system call using Machine learning and testing technique		A	D	H	Kernel
Dixon and Mishra (2013)	Detecting malware through power consumption profile based on time and location		A	D	H	Hardware
Sanz *et al*. (2013a)	Detects malware through extracting several features from the Android Manifest and applied it through machine-learning classifiers	MAMA	A	S	H	App. Package
Moonsamy *et al*. (2013)	A novel pattern mining algorithm to identify a set of contrast permission patterns that aim to detect the malicious application		A	S	H	App. Package
Gascon *et al*. (2013)	Malware detection based on efficient embedding of function call graphs with an explicit feature map inspired by a linear-time graph kernel		A	S		App. Package
Sanz *et al*. (2013b)	An instance-based anomaly detection method that able to detect anomalous malicious applications by analysing the Manifest file of Android applications		A	S		App. Package
Aafer *et al*. (2013)	A robust and lightweight approach for detecting android malware based on different classifier that rely on the semantic information within the byte code of the applications ranging from critical API calls, package level information and some dangerous parameters invoked	DroidAPIMiner	A	S	H	App. Package
Shabtai *et al*. (2012)	Framework for detecting malware on Android mobile using applied Machine learning	Andromaly	A	D	H	All except App. Package
Wei *et al*. (2012)	Determine the intrinsic Android malware domain name resolution communication behavior by extracting the network spatial features of Android apps and used Independent Component Analysis (ICA)		A	D	H	Network Traffic
Wu *et al*. (2012)	Detecting Android Malware using feature-based mechanism for characterizing the Android applications behaviour through the static information from permissions, deployment of components, Intent messages passing and API calls	Droidmat	A	S	H	App. Package

Table 1: Continue

Author	Description	Proposed malware detection system	*Technique	**Approach	***Platform	Data type
Burguera *et al*. (2011)	Detecting malware behaviour by using crowdsourcing system to obtain the traces of applications' behav- ior and analysing the behavior of mobile malware by processing it on server	Crowdroid	A	D	S	Kernel
Portokalidis *et al*. (2009), (2010)	A malware detection architecture where the security checks are applied on remote security servers that host exact replicas of the phones in virtual environments	Paranoid	A	D	S	Kernel
Zhou *et al*. (2012)	A detection that introduce two scheme of detection in which one statically analyse the manifest file from the application package and the second scheme monitors the dynamic loading of the java binary and native code through heuristic analysis	Droid ranger	S/A	D/S	S	App. Package
Dini *et al*. (2013)	Detecting repackaged applications based upon probabilistic automata which computed from the application system call traces	PICARD	SP	D	H	Kernel

*Detection technique: Signature-based (S), Anomaly-based (A), Specification-based (SP), Hybrid (SPA/PA), **Detection approach: Dynamic (D), Static (S)

***Detection platform: Host (H), Server/Cloud(S)

Atzeni *et al.*, 2013; Dixon and Mishra, 2013; Wei *et al.*, 2012; Burguera *et al.*, 2011; Portokalidis *et al.*, 2009; Portokalidis *et al.*, 2010; Dini *et al.*, 2013) were using the dynamic approach. Despite, the advantages it has over static analysis, the dynamic approach tend to produce high generation of captured data that can consume the storage space and the requirement of high computational power to process the data. Besides that, times required to observe the appearance of the malicious activity cannot be define clearly, some malicious activity can be trigger as instant as it being installed to the devices or it might be triggered after minutes, hours or even days. From the reviewed studies only Zhou *et al.* (2012) has used both approaches.

In the platform deployment category, most of the researches done are deploying the Android malware detection system in the host itself. The Android malware detection system developed by Faruki *et al.* (2013), Xu *et al.* (2012), Enck *et al.* (2010), Fuchs *et al.* (2009), Arp *et al.* (2014), Abela *et al.* (2013), Dini *et al.* (2012), Atzeni *et al.* (2013), Dixon and Mishra (2013), Sanz *et al.* (2013a), Moonsamy *et al.* (2013), Aafer *et al.* (2013), Shabtai *et al.* (2012), Wei *et al.* (2012), Wu *et al.* (2012) and Dini *et al.* (2013) were implementing the detection mechanism on the device itself. Host based detection contribute in a faster feedback as the audit source data is processed in the device itself but the amount of data collected during the monitoring will consume the limited storage space provided. To overcome the limited resources a mobile device has, research by Grace *et al.* (2012), Isohara *et al.* (2011), Burguera *et al.* (2011), Portokalidis *et al.* (2009), Portokalidis *et al.* (2010) and Zhou *et al.* (2012) were processing the monitored data on a centralize server or in the cloud infrastructure. This mean the monitored log file from the devices is transferred away to the server side to be process and the result of the detection is send back to the device. Eventhough it's help in reducing the storage space usage and reduced the complex processing in the mobile device, the drawbacks of this process is the exposure of user's confidential data being submitted to the server. Additionally the requirements that need the mobile device to always connect to the internet can cause exhaustion to the battery. On the other hand, studies by Chan *et al.* (2012) and Enck *et al.* (2009b) can be implemented in either host or server side.

The type of audit data source used in the detection can also contribute in improving or downgrading the performance of Android malware detection. Application package that consist of the manifest file, dex classes and byte code is less complex to analyse but it can be difficult to analyse if the dex classes and byte code is obfuscate and encrypted. Study conducted by Faruki *et al.* (2013), Grace *et al.* (2012), Chan *et al.* (2012), Enck *et al.* (2010), Fuchs *et al.* (2009), Arp *et al.* (2014), Sanz *et al.* (2013a), Moonsamy *et al.* (2013), Gascon *et al.* (2013), Sanz *et al.* (2013b), Aafer *et al.* (2013), Wu *et al.* (2012) and Zhou *et al.* (2012) were among the Androidmalware detection system that only analyse the Android application package. The study that only analyse the sensitive data flow is (Enck *et al.*, 2009b). Its monitors confidential data process by application from the application level until the kernel level and network payload but the fact that not all malicious application are processing sensitive information can be a limitation for this approach. Instead of using application package and tracking the flow of sensitive data, Run-time application, kernel level system call and network traffic can be used to reveal the dynamic behaviour of the malware and it is not affected by encryption or obfuscation. However, it can produce a huge amount of data to be process particularly if the information captured is not carefully selected. The studies that only consider Kernel level system call are (Xu *et al.*, 2012; Abela *et al.*, 2013; Atzeni *et al.*, 2013; Burguera *et al.*, 2011; Portokalidis *et al.*, 2009; Portokalidis *et al.*, 2010; Dini *et al.*, 2013) while study done by Wei *et al.* (2012), only audits the network traffic. Only study of Isohara *et al.* (2011) was considering both the Kernel system

Table 2: Advantage and disadvantages of each classification categories

Categories	Advantages	Disadvantages
Techniques		
Signature-based	Detect known attack accurately	Less effective to unknown or new malware
	Using less computational resources	
Anomaly-based	Effectively detect unknown malware	Require more computational resources
		Produce some amount of false alert
Specification-based	Effectively detect unknown malware	Deriving detail specification for SPB is time consuming
		There is a chance the detection will produce false negatives
Analysis		
Static	Quickly detect malware	Able to prevent the malicious application from being installed
		Can be evaded through obfuscation or encryption technique
Dynamic	Obfuscation and encryption	Produce high generation of captured
	technique will not affect the detection	Consume the storage space
		Requires high computational power to process the data
		Times required to observe the appearance of the
		malicious activity cannot be define clearly
Deployment platform		
Host	Faster feedback to the user	Data collected during the monitoring will consume the limited
		storage space
Server/cloud	Reducing the storage space usage	Exposure of user's confidential data
	Reduced the complex processing	Continuous connection to the internet can cause exhaustion
		to the battery
Audit data source		
Application package	Less complex to analyse	Can be evaded through obfuscation or encryption technique
		Not revealing the true behaviour of the malware
Sensitive data flow	Less complex to analyse	Not all malicious application are processing sensitive information
Kernel level	Canreveal the dynamic behaviour	Requires complex analysis
system call	of the malware	Produce a huge amount of data
Network traffic	Canreveal the dynamic behaviour	Complex process
	of the malware	Produce a huge amount of data
Hardware	Canreveal the true behaviour	The abnormality can also be affected by the heavy usage
	of the malware	from the user

call and the data flow. Alternatively, hardware profile performance can help indicate the abnormal behaviour of the device due to the malicious activity, yet the abnormal performance can also be affected by the heavy usage from the user. This category of hardware profile audit data source is only use by the study of Dixon and Mishra (2013). Still, there are two Android malware detection systemsthat considering almost all the audit source except the application package, which are the study of Dini *et al.* (2012) and Shabtai *et al.* (2012). All the comparison between the categories of the classification is summarized in Table 2.

Albeit the several studies on Android malware detection, none of them won through to be the most accurate and reliable detection approach, each of the techniques have it benefits and drawbacks. As a result a combination of two or more categories from the classification in Fig. 2, is needed to compensate between the benefit and limitation of each detection techniques, detection approach, and detection deployment platform and detection audit data source. Finally to summarize and refine the classification of Android malware detection system, this study proposed a new perspective of Android malware detection system taxonomy as depicted in Fig. 2.

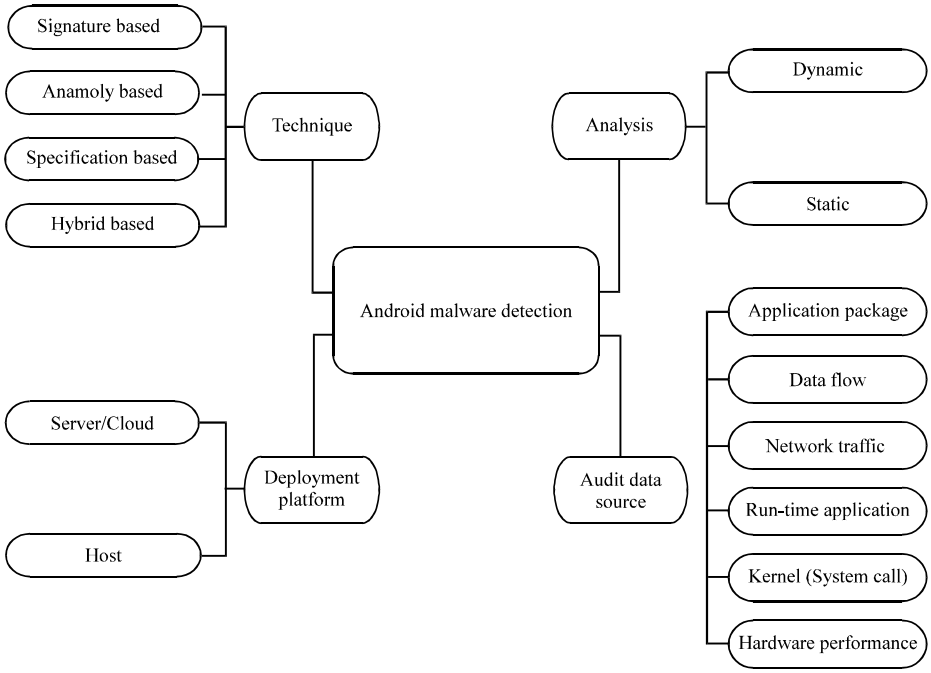

Fig. 2: Android malware detection system taxonomy

CONCLUSION

The rapid evolution of mobile devices technology has also increased the number of mobile malware in the application market particularly when Android OS is widely adopted in the mobile devices. Even though a large number of research has been done in defending computer from malware infection, adopting the solution done in desktop environment into mobile devices is not that easy, multiple factor have to be consider before the assimilation can be done. This study has comprehensively reviewed and classifies existing Android malware detection system. From the review, we have introduced an Android malware detection taxonomy that classifies the detection system into detection techniques, approaches, deployment platform and the audit data source. Further analysis shows that each category used in the detection systemhas it superiority and limitations; hence,only the right combination of the classification can improve the reliability and the effectiveness of Android malware detection system. In general Android malware detection needs to be able to detect known and unknown malware especially the zero-day malware without generating a high false alarm. Moreover the Android malware detection system must take the consideration of the limited resources the mobile devices provide. This study is the preliminary work for finding a more robust, reliable and effective Android malware detection system.

ACKNOWLEDGEMENT

The authors would like to express the appreciationto InforsNet Group of Universiti Teknikal Malaysia Melaka (UTeM) and Ministry of Education Malaysia (MoE) for their invaluable supports in encouraging the authors to publish this study. This study was supported by the MoE under Grants LRGS/TD/2011/UKM/ICT/02/02.

REFERENCES

Aafer, Y., W. Du and H. Yin, 2013. DroidAPIMiner: Mining API-level features for robust malware detection in Android. Proceedings of the International Conference on Security and Privacy in Communication Networks, September 25-28, 2013, Sydney, NSW, Australia, pp: 86-103.

Abela, K.J., J.R.D. Alas, D.K. Angeles, R.J. Tolentino and M.A. Gomez, 2013. Automated malware detection for android AMDA. Proceedings of the 2nd International Conference on Cyber Security, Cyber Peacefare and Digital Forensic, March 4-6, 2013, Kuala Lumpur, Malaysia, pp: 180-188.

Armando, A., A. Merlo and L. Verderame, 2012. Security issues in the android cross-layer architecture. ArXiv Preprint arXiv:1209.0687, September 4, 2012.

Arp, D., M. Spreitzenbarth, M. Hubner, H. Gascon and K. Rieck, 2014. Drebin: Effective and explainable detection of android malware in your pocket. Proceedings of the Network and Distributed System Security (NDSS) Symposium, February 23-26, 2014, San Diego, California.

Atzeni, S., M. Dimjasevic, C. Schreiner and Y. Wang, 2013. Automatic malware detection for android. Proceedings of the 1st University of Utah CS5350/CS6350 Conference on Machine Learning, December 3-12, 2013, Salt Lake City, Utah.

Burguera, I., U. Zurutuza and S. Nadjm-Tehrani, 2011. Crowdroid: Behavior-based malware detection system for android. Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, October 17-21, 2011, Chicago, IL., USA., pp: 15-26.

Castillo, C.A., 2013. Android malware past, present and future. White Paper, Mcafee, January, 2013. http://www.mcafee.com/us/resources/white-papers/wp-Android-malware-past-present-future.pdf.

Chan, P.P.F., L.C.K. Hui and S.M. Yiu, 2012. DroidChecker: Analyzing android applications for capability leak. Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks, April 16-18, 2012, Tucson, AZ., USA., pp: 125-136.

Dini, G., F. Martinelli, A. Saracino and D. Sgandurra, 2012. MADAM: A multi-level anomaly detector for Android malware. Proceedings of the 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security: Computer Network Security, October 17-19, 2012, St. Petersburg, Russia, pp: 240-253.

Dini, G., F. Martinelli, A. Saracino and D. Sgandurra, 2013. Probabilistic contract compliance for mobile applications. Proceedings of the 8th International Conference on Availability, Reliability and Security, September 2-6, 2013, Regensburg, pp: 599-606.

Distler, D., 2013. Malware analysis: An introduction. SANS Institute InfoSec Reading Room, July, 2013.https://www.sans.org/reading-room/whitepapers/malicious/malware-analysis-introduction-2103.

Dixon, B. and S. Mishra, 2013. Power based malicious code detection techniques for smartphones. Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), July 16-18, 2013, Melbourne, VIC., pp: 142-149.

Enck, W., M. Ongtang and P. McDaniel, 2009a. Understanding android security. IEEE Security Privacy, 7: 50-57.

Enck, W., M. Ongtang and P. McDaniel, 2009b. On lightweight mobile phone application certification. Proceedings of the 16th ACM Conference on Computer and Communications Security, November 9-13, 2009, Chicago, Illinois, USA., pp: 235-245.

Enck, W., P. Gilbert, B.G. Chun, L.P. Cox, J. Jung, P. McDaniel and A.N. Sheth, 2010. TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones. Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation, Volume 10, October 4-6, 2010, Vancouver, BC., Canada, pp: 255-270.

Farmer, D. and W. Venema, 2004. Forensic Discovery. Addison Wesley Professional, New York, USA., ISBN: 9780321703255, Pages: 217.

Faruki, P., V. Ganmoor, V. Laxmi, M.S. Gaur and A. Bharmal, 2013. AndroSimilar: Robust statistical feature signature for android malware detection. Proceedings of the 6th International Conference on Security of Information and Networks, November 26-28, 2013, Aksaray, Turkey, pp: 152-159.

Felt, A.P., M. Finifter, E. Chin, S. Hanna and D. Wagner, 2011. A survey of mobile malware in the wild. Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, October 17-21, 2011, Chicago, Illinois, USA., pp: 3-14.

Felt, A.P., E. Ha, S. Egelman, A. Haney, E. Chin and D. Wagner, 2012. Android permissions: User attention, comprehension and behaviour. Proceedings of the 8th Symposium on Usable Privacy and Security, July 11-13, 2012, Washington, DC., USA., pp: 3.

Fuchs, A.P., A. Chaudhuri and J.S. Foster, 2009. SCAndroid: Automated security certification of android applications: Technical Report CS-TR-4991. Department of Computer Science, University of Maryland, College Park, November 2009.

Funk, C. and M. Garnaeva, 2013. Kaspersky security bulletin 2013. Overall statistics for 2013. Kaspersky Lab Global Research. http://www.securelist.com/en/analysis/204792318/Kaspersky_Security_Bulletin_2013_Overall_statistics_for_2013.

Gascon, H., F. Yamaguchi, D. Arp and K. Rieck, 2013. Structural detection of android malware using embedded call graphs. Proceedings of the 2013 ACM Workshop on Artificial intelligence and Security, November 4, 2013, Berlin, Germany, pp: 45-54.

Grace, M., Y. Zhou, Q. Zhang, S. Zou and X. Jiang, 2012. Riskranker: Scalable and accurate zero-day android malware detection. Proceedings of the 10th International Conference on Mobile Systems, Applications and Services, June 25-29, 2012, Low Wood Bay, Lake District, UK., pp: 281-294.

ITU, 2013. The World in 2013, ICT facts and figures. International Telecommunication Union. http://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx.

Isohara, T., K. Takemori and A. Kubota, 2011. Kernel-based behavior analysis for android malware detection. Proceedings of the 7th Conference on Computational Intelligence and Security, December 3-4, 2011, Hainan, pp: 1011-1015.

Jiang, X., 2012. An evaluation of the application (App) verification service in android 4.2. Department of Computer Science, NC State University, December 10, 2012. http://www.cs.ncsu.edu/faculty/jiang/appverify/.

Juniper Networks, 2011. Mobile threats report 2011. Juniper Network Mobile Threat Center, February 2012. http://www.juniper.net/us/en/local/pdf/additional-resources/jnpr-2011-mobile-threats-report.pdf.

Ko, C., G. Fink and K. Levitt, 1994. Automated detection of vulnerabilities in privileged programs by execution monitoring. Proceedings of the 10th Annual Computer Security Applications Conference, December 5-9, 1994. Orlando, Florida, pp: 134-144.

Kumar, V., J. Srivastava and A. Lazarevic, 2005. Managing Cyber Threats: Issues, Approaches and Challenges. 1st Edn., Vol. 5, Springer, New York, ISBN: 9780387242309, pp: 330.

La Polla, M., F. Martinelli and D. Sgandurra, 2013. A survey on security for mobile devices. IEEE Communi. Surveys Tutorials, 15: 446-471.

Liao, H.J., C.H.R. Lin, Y.C. Lin and K.Y. Tung, 2013. Intrusion detection system: A comprehensive review. J. Network Comput. Applic., 36: 16-24.

Moonsamy, V., J. Rong, S. Liu, G. Li and L. Batten, 2013. Contrasting Permission Patterns between Clean and Malicious Android Applications. In: Security and Privacy in Communication Networks, Zia, T., A. Zomaya, V. Varadharajan and M. Mao (Eds.). Springer, USA., ISBN: 978-3-319-04282-4, pp: 69-85.

Okazaki, Y., I. Sato and S. Goto, 2002. A new intrusion detection method based on process profiling. Proceedings of the Symposium on Applications and the Internet, January 28-February 1, 2002, Nara, Japan, pp: 82-90.

Pieterse, H. and M.S. Olivier, 2012. Android botnets on the rise: Trends and characteristics. Proceedings of the IEEE Conference on Information Security for South Africa, August 15-17, 2012, Johannesburg, Gauteng, pp: 1-5.

Portokalidis, G., P. Homburg, K. Anagnostakis and H. Bos, 2010. Paranoid android: Versatile protection for smartphones. Proceedings of the 26th Annual Computer Security Applications Conference, December 6-10, 2010, Austin, TX., USA., pp: 347-356.

Portokalidis, G., P. Homburg, N. FitzRoy-Dale, K. Anagnostakis and H. Bos, 2009. Protecting smart phones by means of execution replication. Technical Report IR-CS-54. http://www.cs.vu.nl/~herbertb/papers/pa09.pdf.

Preda, M.D., M. Christodorescu, S. Jha and S. Debray, 2007. A semantics-based approach to malware detection. Proceedings of the 34th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Volume 42, January 17-19, 2007, Nice, France, pp: 377-388.

Robiah, Y., S.S. Rahayu, M.M. Zaki, S. Shahrin, M.A. Faizal and R. Marliza, 2009. A new generic taxonomy on hybrid malware detection technique. Int. J. Comput. Sci. Inform. Secur., 5: 56-61.

Sanz, B., I. Santos, X. Ugarte-Pedrero, C. Laorden, J. Nieves and P.G. Bringas, 2013a. Instance-based anomaly method for android malware detection. Proceedings of the 10th International Conference on Security and Cryptography, July 29-31, 2013, Reykjavik, Iceland, pp: 387-394.

Sanz, B., I. Santos, C. Laorden, X. Ugarte-Pedrero, J. Nieves, P.G. Bringas and G.A. Maranon, 2013b. MAMA: Manifest analysis for malware detection in android. Cybernetics Syst., 44: 469-488.

Shabtai, A., Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev and C. Glezer, 2010. Google Android: A comprehensive security assessment. IEEE Security Privacy, 8: 35-44.

Shabtai, A., U. Kanonov, Y. Elovici, C. Glezer and Y. Weiss, 2012. Andromaly: A behavioral malware detection framework for Android devices. J. Intell. Inform. Syst., 38: 161-190.

Shaerpour, K., A. Dehghantanha and R. Mahmod, 2013. Trends in android malware detection. J. Digital Forensics, Security Law, Vol. 8.

Sundaram, A., 1996. An introduction to intrusion detection. Magazine Crossroads, 2: 3-7.

Van der Meulen, R. and J. Rivera, 2012. Gartner says smartphone sales accounted for 55 percent of overall mobile phone sales in third quarter of 2013. Gartner, Barcelona, Spain, November 14, 2013, http://www.gartner.com/newsroom/id/2623415.

Wei, T.E., C.H. Mao, A.B. Jeng, H.M. Lee, H.T. Wang and D.J. Wu, 2012. Android Malware detection via a latent network behavior analysis. Proceedings of the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, June 25-27, 2012, Liverpool, pp: 1251-1258.

Wu, D.J., C.H. Mao, T.E. Wei, H.M. Lee and K.P. Wu, 2012. DroidMat: Android malware detection through manifest and API calls tracing. Proceedings of the 7th Asia Joint Conference on Information Security, August 9-10, 2012, Tokyo, pp: 62-69.

Xu, R., H. Saidi and R. Anderson, 2012. Aurasium: Practical policy enforcement for android applications. Proceedings of the 21st USENIX Conference on Security Symposium, August 8-10, 2012, Bellevue, WA., USA., pp: 27.

Zhou, Y. and X. Jiang, 2012. Dissecting android malware: Characterization and evolution. Proceedings of the IEEE Symposium on Security and Privacy, May 20-23, 2012, San Francisco, California, pp: 95-109.

Zhou, Y., Z. Wang, W. Zhou and X. Jiang, 2012. Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets. Proceedings of the 19th Annual Network and Distributed System Security Symposium February 5-8, 2012, San Diego, California, USA.