ISSN:0975-9646



# A Survey on Android Malwares and their Detection Mechanisms

Nancy, Dr. Deepak Sharma

*Department of CSE, Kurukshetra Institute of Technology and Management*
*Kurukshetra, India*

***Abstract*— Android smartphones now a days are the most used smartphones with more than 80% of the users using Android. This is due to the presence of feature rich apps present in Google Play Store. Such popularity of Android platform has come hand in hand with increase in number of malwares targeting Android. Since 2010 Android malwares have been on the high and more than 98% of mobile malwares target Android. Threats posed by malwares include leaking of private information, financial loss to users, system damage etc. In this paper we highlight why Android is the most targeted mobile platform for the malware developers and how malwares propagate into the Android system. We also discuss what are the threats posed by them, what detection techniques have been proposed in the literature for detecting them, and what are the existing research gaps in detection mechanisms.**

***Keywords*— Android Malware, Android Security, Mobile Attacks**

## I. INTRODUCTION

Apart from conventional services like phone calls, SMS, MMS etc. smartphones have become ubiquitous due to the presence of various feature rich apps providing services like online banking, social networking, enterprise applications, numerous types of games, location based services, availability of Internet etc. Smartphones have swiftly emerged as an attractive gadgets with powerful computing capabilities; which at current times are more powerful than earlier generation of personal computers (PCs). In the past few years there has been a tremendous growth in the sales of smart phones. According to [1] the number of smartphone users has increased from 1.5 billion in 2014 to expected 2 billion users by the end of 2016, registering an increase of about 33%. Moreover Smartphones and Tablets have even outperformed the sales of traditional PCs (desktops and notebooks) [2]. Google's Android has been the bestseller amongst others like iOS, Symbian, Windows etc. since around 83 % of the smartphones sold in 2014 and till second quarter of 2015 were Android based [3] and Google's App Store being the largest App Store with 2.2 million apps while Apple's App Store is at second position with 2 million apps [4]. Such rapid increase in popularity of smartphones and its worldwide user acceptance has come hand in hand with analogous rise in attacks targeting popular mobile platforms. Smartphones now a days pose greater security and privacy threat to users than traditional desktop systems [5] because of the presence of numerous sensors incorporated in the device which may leak sensitive information regarding the location of the user or the information stored by user (users can store their authentication credentials in their device), may record audio or video from its surroundings etc. Furthermore smartphones and PC's are similar in the sense that both need an operating system, so malicious attacks of worms, viruses and Trojans etc. which have been common on desktops are applicable to smartphones as well. Hence it has been easier for malware developers to move from desktop environment to mobile devices.

792 mobile malware samples were collected by McAfee Labs by 2011 and the number increased to around 8,000 during first quarter of 2012. Around 800 new samples every month were found in 2011 and this number increased to 6300 samples being detected every month in the year 2012 [6]. 1, 45,000 new malwares were detected by the end of 2013 out of which 98% targeted Android phones. By the end of second quarter of 2015 nearly 1 million new samples have been found targeting Android which accounts for nearly 6,100 malwares per day or new malware instance every 14 seconds which clearly shows that malware developers are continuing their implacable development of new malwares targeting mobile platforms. Android at present is the prime target for attackers because of the presence of three factors [7]:

1) Motive: According to [3] Android is winning the race for the top mobile platform. Prospect of targeting large number of users community as now around 80% of the smartphone users are Android based, contributes a strong motive to write malwares for this platform.
2) Means: Symbian was the most popular mobile platform before Android but the only way to spread malware in Symbian was through Bluetooth and therefore this required physical propinquity of the device with its Bluetooth on. But Android provide a simpler solution to spread malware i.e. through apps as there are so many third party app markets in addition to official Play Store.
3) Opportunity: Although Apple also provides app store through which malwares can propagate but Android is open source whereas Apple OS is closed in nature. Moreover Apple play store apply more rigorous process than Android in reviewing apps. Therefore Android is the most targeted platform for the mobile malware developers.

## II. MALWARE PROPAGATION

Different ways through which malwares can propagate in Android are:

1) Repackaging: Malware developers first download any popular app, disassemble that app (i.e. generating the source code written in Java), insert

their own code having malicious payload within the original code, reassemble the app and redistribute that app in official or third party app markets. According to [7], out of 1260 samples 1083 of them (86%) are repackaged.

2) Update Attacks: Repackaging technique includes malicious payload within the original app but that is easier to detect by analysing the source code. To evade detection malware developers instead of including malicious payload within the app, they include only an update component which downloads the malicious payload at run time after the app is installed on the device. Hence scanning the source code will not be able to detect the malware as initially there is no malicious code within the app.
3) Drive-by-Downloads: This technique employs traditional drive-by-downloads to Android devices as well in which users are enticed to download interesting or attractive apps. For example, GGTracker malware has in-app advertisement library. After clicking on that advertisement link user is redirected to a website which displays the message to download an app which can save battery of the device. However that downloaded app is actually a malware which subscribes to premium rate services without user's knowledge

## III. Malware Threats

Once smartphones are infected by malwares, they can cause damage / interruption in the services to the users like [8]:

1) System Damage: Some malwares are successful in bringing the device to hang state where the mobile doesn't operate normally. They may even block calling functionality in mobile devices or can root the Android OS. Battery draining is also one of the consequences.
2) Financial Loss: Sending SMS/MMS messages to premium rate numbers or dialling phone call to premium rate numbers without users' knowledge in the background incurs financial loss to the user of the mobile device.
3) Information Leakage: Some malwares can enable attacker with the capability to browse through private information like SMS/MMS, contact details, call logs, emails etc.
4) Remote Control: Smartphones can even behave as "bots" i.e. like robots under the control of a remote server. After infection device can receive commands from the server and perform the corresponding action. The action can be to send spam mails or to send SMS or to root the phone etc.

## IV. Detection Mechanisms

Although Google Play Store uses its anti-virus named "Bouncer" to detect malicious apps but still many malicious apps have been found in play store since 2010. Other anti-virus are also incapable of detecting malwares with high accuracy as shown in the table I. [7]

TABLE I
DETECTION ACCURACY OF ANTI-VIRUSES

| Anti-Virus | Detection Accuracy |
|---|---|
| AVG | 54 % |
| Lookout | 79 % |
| Norton | 20 % |
| TrendMicro | 76 % |

Therefore the researchers felt the need of developing different mechanisms to detect Android malwares with high accuracy. Solutions proposed so far lie either in two categories: Static Analysis and Dynamic Analysis.

### A. Static Analysis

Static analysis is a quick as well as inexpensive approach which aims at finding malicious characteristics or bad code segments in an application without executing them. These techniques are generally used in a preliminary analysis when suspicious applications are first evaluated to detect any security threats. These type of detection mechanisms make use of Dalvik decompiler to generate java source code from the android app (apk file) and then analyse that code to look for suspicious behaviour in terms of dangerous permissions used. Features used in detection include: Permissions, Java code, Intent Filters, Hardware components etc. where permissions and java code are the most used features compared to others [9]. So we will discuss about these features.

Permissions which are defined in the Android Manifest file of the application, are the most used feature in static detection of malwares. Whenever any malicious app has to do some dangerous activity it must have permission to do so and it should be defined in the Android Manifest file. For example if any game application is having permission of SEND_SMS, then it may be malicious. Similarly if any application has permission patterns of INTERNET and READ_PHONE_STATE, then there may be chances that application might be leaking device information to the remote server. All such permissions must be requested by the application in its manifest file only then Android OS will allow application to use that required component. Therefore permissions are the most used static feature in Android malware detection.

Java Code is another used static feature used in malware detection. All Android apps are written in Java programming language and are then compiled to a format known as Dalvik executable. Code includes all the API calls which are made by the application and the researchers look for finding these malicious APIs or malicious keywords defined in the code. For example *getContentResolver().delete()* is one such malicious API which can delete SMS or even delete a file. *getLineNumber* API is used by malwares for phone number leakage. So the focus is to look out for these malicious APIs defined in the code to check whether application is malicious or not.

Summary of few static solutions proposed have been discussed in the table II below:

TABLE II
SUMMARY OF STATIC SOLUTIONS

| Name of Model | Contribution |
|---|---|
| Adrisk [10] | Study the existing in app ad-libraries and evaluate potential risks from them |
| Permissions Pattern [11] | Identify permission set that can distinguish between malicious apps and normal apps |
| AndroGuard [12] | To detect repackaged apps from official markets by finding similarities between apps |
| AndroSimilar [13] | Detect unknown new malwares which employed obfuscation methods like string encryption |
| DroidAnalyzer [14] | Detect malwares having root exploits within their java code. |

Although Static Analysis is easy to perform; the app is not being executed and simply permissions and code is analysed but it has a major limitation. There is low accuracy in detecting mobile malwares as many of them evade static analysis by employing techniques like update attacks. In update attacks malwares doesn't include any malicious component initially in the app but downloads malicious code at run time after its installation on the device. Hence static analysis is unable to detect these type of malwares.

### *B. Dynamic Analysis*

To overcome this limitation of static analysis, dynamic solutions were proposed in which app's runtime behaviour is observed by executing the app either on emulator or actual smartphone. Malwares with update attacks can evade static analysis but they are more likely to be detected when their run time dynamic behaviour is observed. Two features are primarily used in monitoring dynamic behaviour of apps which are system calls and network behaviour [9].

System call analysis has been used extensively in desktop malware detection. Since Android is also a OS hence it also produces system calls for any activity performed and hence system calls can be used for looking for signs of suspicious behaviour. Considering the example again where any gaming app is generating system call of sending SMS, it is a sign of suspicious behaviour.

Network Traffic has also been used in intrusion detection in the past and can be used in Android malware detection as well in which traffic features are observed for normal traffic and malicious apps traffic. Traffic files of both are compared and distinguishing features are determined on which classification of malicious or normal apps can be done. This type of analysis is highly accurate but it can be used only for a subset of malwares which produce network traffic. Those malwares which don't have any network connectivity will not produce any network traffic and hence such analysis will not work for these type of malwares.

Summary of few dynamic solutions proposed have been discussed in the table III below:

TABLE III
SUMMARY OF DYNAMIC SOLUTIONS

| Name of Model | Contribution |
|---|---|
| SCSDroid [15] | Analysing thread grained system call sequences rather than process grained system calls to detect repackaged malwares. |
| CrowDroid [16] | Analysing system calls of apps and apply clustering to divide apps in normal or malicious category. |
| DroidRanger [17] | Two phase analysis: behavioural permission footprint (static) and heuristics based filtering of system calls (dynamic) |
| RiskRanker [18] | Two order risk analysis: first static permission based, second based on dynamic dalvik code loading |
| Network Traffic Based [19] | Find distinguishing network traffic features between malicious apps traffic and normal apps traffic. |

Although these dynamic methods overcome the limitations of low accuracy of static solutions, but they still have their own limitations like: consume device's limited resources while capturing run time behaviour of app on actual smartphone. If emulator is used then not all malwares will generate run time behaviour as some malwares wait for system events like receiving a SMS or phone call to activate their malicious payloads but such events are not possible in emulator. Anti-emulation techniques like Sandbox, delaying of malware execution etc. can help in evade dynamic analysis.

## V. RESEARCH GAPS

We have seen both types of detection mechanisms along with their advantages and disadvantages. There are lots of research gaps that have been found which are yet to work upon like:

1) Low accuracy of static solutions as malwares with update attack easily evade detection. Merely analysing their permissions or java code is not enough to detect them as malwares.
2) Dynamic Analysis consumes limited resources of the mobile device like CPU, memory, battery etc. if run time behaviour is observed on the actual device.
3) Using emulator for run time behaviour is also used for a subset of malwares which don't depend upon any system event to get activated. Hence emulator cannot be used to capture behaviour of every type of malware.
4) Most of the techniques lack in detecting zero day / unknown malwares. This is the reason behind large number of malicious apps still found in official or third party app stores.

## VI. CONCLUSION

In this paper we have discussed about Android malwares; how they propagate and what are the threats posed by them. We have also discussed different types of solutions proposed in the literature for Android malware detection. Static solutions analyse permissions and code defined in the app without executing the app. Dynamic solutions monitor run time behaviour of apps by executing them on emulator or actual smartphone. But still there are some research gaps which have been addressed in the paper. There is a need to develop more advanced solutions which can detect zero day or unknown malwares with high accuracy.

## REFERENCES


[1] Statista, "Number of smartphone users worldwide from 2014 to 2019" [Online]. Available: http://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

[2] Chris Taylor, "Smartphone Sales Overtake PCs for the First Time [Online]. Available. http://mashable.com/2014/02/03/smartphone-sales-overtake-pcs/, January 10, 2014

[3] IDC, "Smartphone OS Market Share, 2015 Q2", [Online]. Available. http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[4] Statista, "Number of apps available in leading app stores as of June 2016" [Online]. Available. http://www.statista.com/statistics/276623/number-of-apps-available-in- leading-app-stores/

[5] A. Makhlouf and N. Boudriga, "Intrusion and anomaly detection in wireless networks," in Handbook of Research on Wireless Security, Information Science Publishing, 2008.

[6] Kaspersky, "99% of all mobile threats target Android devices" [Online]. Available: http://www.kaspersky.com/about/news/virus/2013/99_of_all_mobile_threats_target_Android_devices

[7] Y. Zhou., and X. Jiang, "Dissecting Android Malware: Characterization and Evolution," In Proceedings of the 33rd IEEE Symposium on Security and Privacy (2012), IEEE Oakland 2012.

[8] S. Peng, S. Yu, and A. Yang, "Smartphone malware and its propagation modeling: A survey", IEEE Communications Surveys Tutorials vol. 99, pp.1-17, 2013.

[9] A Feizollah, N.B. Anuar, R. Salleh, A. W. A. Wahab, "A review on feature selection in mobile malware detection", Digital Investigation, vol. 13, June 2015, pp. 22-37.

[10] M. Grace, W. Zhou, X. Jiang A. Sadheghi "Unsafe Exposure Analysis of Mobile In-App Advertisement", In Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks (2012), ACM WiSec 2012.

[11] V. Moonsamy, J. Rong, S. Liu "Mining permission patterns for contrasting clean and malicious android applications" Future Generation Computer Systems, vol. 36, September 2013, pp.122-132.

[12] BlackHat, Reverse Engineering with Androguard, [Online]. Available: https://code.google.com/androguard

[13] P. Faruki, V. Ganmoor, V. Laxmi, M. S. Gaur, A. Bharmal, "AndroSimilar: Robust Statistical Feature Signature for Android Malware Detection" In Proceedings of the 6th International Conference on Security of Information and Networks SIN '13, New York, USA, 2013.

[14] S.H. Seo, A. Gupta, A.M. Sallam, E. Bertino, K. Yim,, "Detecting mobile malware threats to homeland security through static analysis,", Journal of Network and Computer Applications, vol. 38, pp.43-53.

[15] Y. Lin, Y. Lai, C. Chen, H. Tsai, "Identifying android malicious repackaged applications by thread-grained system call sequences", Computers & Security, vol. 39, pp.340-350.

[16] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani. "Crowdroid: Behavior-Based Malware Detection System for Android". In Proceedings of the 1st Workshop on Security and Privacy in Smartphones and Mobile Devices CCS-SPSM'11, 2011.

[17] Y. Zhou, Z. Wang, W. Zhou, X. Jiang, "Hey You, Get off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets." In Proceedings of the 19th Annual Symposium on Network and Distributed System Security (2012), NDSS 2012

[18] M. Grace, Y. Zhou, Q. Zhang, S. Zou, X. Jiang, "RiskRanker: Scalable and Accurate Zero-day Android Malware Detection", In 10th International Conference on Mobile Systems, Applications and Services, June 2012

[19] A. Arora, S.Garg, S.K. Peddoju, "Malware Detection using Network Traffic Analysis in Android based Mobile devices", In 8th International Conference on NGMAST, Oxford UK, 2014.