Survey of Mobile Malware Analysis, Detection Techniques and Tool

Conference Paper · November 2018

DOI: 10.1109/IEMCON.2018.8614895

2 authors, including:

Nana Kwame Gyamfi

23 PUBLICATIONS **20** CITATIONS

SEE PROFILE

Survey of Mobile Malware Analysis, Detection Techniques and Tool

Nana Kwame Gyamfi
Computer Science Dept.
Kumasi Technical University
Kumasi, Ghana
*nkgyamfi@st.ug.edu.gh

Dr. Ebenezer Owusu
Department of Computer Science
University of Ghana-Legon
Legon, Ghana
*ebeowusu@ug.edu.gh

Abstract

The rapid increase in the use of smartphones, has contributed to the increase in mobile attackers. In most situations deceitful applications are infected with malicious contents to cause harm to both the hardware and the software. These malicious programs or malware are usually designed to disrupt or gather information from the device. By attempts to curtail these problems various techniques are proposed. This paper attempts to analyze the most popular and recent techniques and suggests which is better.

Keywords: Mobile phone; mobile malware; static detection; dynamic detection; hybrid detection.

1. Introduction

Smartphone usage has expanded exponentially and it is continuously transforming into a cutting edge device [20]. Its popularity has made it attractive to attackers. Modern smartphones are more advanced; they are used in businesses transactions and for saving individual data [19]. In effect, this has made them helpless against malware attacks. Malware developers believe that it is easy to transmit attackers to mobile devices because mobile users do not usually have time to analyze applications that are downloaded from the app-stores and websites.

Mobile malware is a malicious software that is designed with specific targets for mobile devices [18]. It first emerged as early as 2004 for Symbian OS [26] and now it has gained exponential growth along with the popularity of smartphones. Figure 1 shows the mobile malware growth since 2010 up to date.

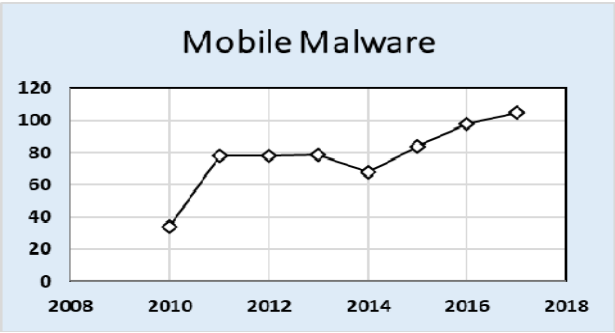

Fig. 1: Mobile Malware Growth

The mobile malware program gets itself installed while accessing the internet with mobile device and then perform functions without user's permission or knowledge. There are many ways for which they are distributed among which are the internet via a mobile browser, downloads and installation via device messaging functions. The mobile malware can be broadly classified into types [13].

The malware is getting more curved with programs that worked outside of anyone's ability to see the client gadgets, covering themselves and lying in sit tight for specific practices like a web based saving money session to strike [7]. Covered strategies can execute absolutely imperceptible to the customer and run executable or contact boot access for new bearings. These actions taken by the reviewing malware lends itself to three threat classifications namely, *financial threat*, example spyware and key loggers, *the masked threat*, example trojan and rootkit and the *contagious threat* such as virus and worms.

There are several mobile malware but they are collectively grouped into phishing Apps, trojans and viruses and spyware.

Phishing Apps are closer to desktop computing where attackers develop applications that resemble genuine administrations, yet are proposed to take sensitive data and accreditations to perform money related extortion. One such case is a fake security application of Tagged, Twitter and Tango, which guaranteed to secure the client's record, yet steal client's data for identity fraud [5].

Trojan is a harmful bit of programming that hides itself and acts as an honest to goodness program to take unapproved control of the device. Trojans do not self-replicate, rather through client communication, for example, downloading a document from the web [8]. They take the client's confidential data without their insight. Unlike Trojans, viruses attach themselves to executable files and it's self-reproducing. Virus infection starts with one gadget, then onto the next [1] and [2].

Spyware for the most part assembles classified data covertly about the cell phone clients and go along this information to a third party. Now and again, these might be promoters or marketing data firms alluded to as "adware" [3]. Spyware uses the loss' flexible relationship to exchange singular information such as contacts, message affinities, program history and customer's inclinations to downloads [6]. Likewise, it assembles data, such as OS version, product ID, IMEI number and global mobile subscriber identity number which can be used for future strikes

The remaining portions of this paper is outlined as follows: Section 2 presents the techniques of malware detection, Section 3 discusses the performance and analysis of static and dynamic techniques, Section 4 discusses the hybrid detection technique and Section 5 discusses the conclusions of the study.

2. Malware detection techniques

Mobile malware detection techniques are grouped into static, dynamic and hybrid but each technique comes with strengths and challenges.

2.1 Static analysis detection technique

This technique is to analyze programs without executing it. Amid static examination, the application is separated by utilizing reverse engineering tools and systems in order to re-construct the source code and calculations that the application is made. Static examination should be possible through program analyzer, debugger and disassembler.
Different static methods are used and they are known as follows:

2.1.1 Signature based detection technique

This method is also known as pattern matching or fingerprinting technique. Here, a signature as a bit of sequence is infused into the application program by malware writers, which extraordinarily recognizes a specific malware [15]. To recognize a malware in the code, the malware indicator has to scan for a formerly determined signature in the code.

2.1.2 Heuristic detection technique

This strategy is otherwise called proactive procedure. This strategy is like the signature based system, however, as opposed to hunting down a specific signature in the code, the malware indicator now scans for the commands or instructions that are absent in the application program [16], [12]. The outcome is that, it turns out to be less difficult to recognize new variations of malware that had not yet been found. Listed below are some techniques for heuristic analysis:

File based heuristic analysis: This strategy is also known as file analysis. With this system, the record is broken down profoundly by its substance, reason and goal. On the off chance that the record contains commands to erase or hurt other document, it is considered as malicious [4].

Weight base heuristic analysis: This is the much antiquated system where every application is weighted by the risk it might have. In the event that the weighted esteem surpasses the predefined edge esteem, the application contains vindictive code.

Ruled based heuristic analysis: This analyzer extricates the guidelines characterizing the application. These principles are coordinated with the already characterizes rules. On the off chance that the principles are confounded, the application contains malware.

Generic signature analysis: In this signature, malware with different behaviors, but belonging to the same cohort are detected. This technique previously defines an antivirus definition, to discover new variants of malware.

2.1.3 Static Analysis tools

These tools are used in a preliminary analysis, when suspicious applications are first evaluated to detect any security threats. Examples of the tools are discussed as follows:

IDA pro: The device is utilized to extricate the system calls made by the application and after that go to the centroid machine to perform irregularity identification and order the application in view of their malicious activities. Figure 2 shows its process in brief.

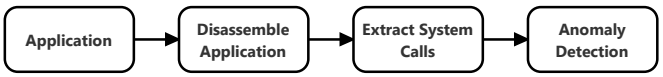

Figure 2: IDA pro process

*PiOS***:** The *PiOS* is utilized for static analysis to check if an application has access sensitives and its capacity to transmit over the network. PiOS first makes a control stream chart from the application binaries earlier before it functions.

UPX: This is a compact, elite executable packer for a few diverse executable configurations. It accomplishes with great compression ratio and offers quick decompression.

*ProcDump***:** The primary design is to screen application for CPU spikes and produce a crash dump, which can be utilized by engineers and heads to decide the reason for the spike. ProcDump additionally incorporates hung window observing and unhandled special case checking that can produce dumps in view of the estimations of framework execution counters. Likewise, it can fill in as a general procedure dump utility that can be installed into different contents.

2.2 Dynamic analysis detection technique

The way toward examining the conduct or the activities performed by the application while it is executing is called dynamic analysis [13]. Dynamic examination should be done through observing function calls, tracking the data stream, breaking down function parameters and tracing direction. For the most part, a virtual machine or sandbox is utilized in this investigation where the questioned application is typically kept running in a virtual domain. On the off chance that the application gets out of hand, it is typically arranged as noxious. This type of detection also come with many forms.

2.2.1 Anomaly Based Detection

[25] Proposed a tool that discover the conduct of applications dynamically. The tool was utilized to conjure

the application for subtle elements (SourceForge.net, 2016). The crowdsourcing application, which is introduced on the gadget, makes a log and sends it to a remote server. Log documents may incorporate the accompanying itemized gadget information and system calls.

[23] Proposed an Android malware recognition framework called andromly. This application ceaselessly screens the state of the gadget at battery level, CPU utilizations and so forth while it is running and afterward apply the machine learning algorithm to distinct amongst malignant and benign applications. The solution can distinguish ceaseless assaults and show the report to client.

2.2.2 Emulation based detection

[21] Proposed mobile dynamic analysis platform called DroidScope which is based on virtual machine introspection. Its main operation is to monitor the whole operating system. Android Application Sandbox is an example of such system as proposed by [22].

2.2.3 Taint Analysis

TaintDroid is another dynamic detection technique. This technique gives system-wide data stream tracking for Android mobile. It can track numerous wellsprings of delicate information such as GPS, camera, and microphone and recognize the information spillage in outsider engineer applications. It tracks and label touchy information from the cell phone.

2.2.4 Dynamic Analysis Tools

These tools that dynamically observe the behavior of mobile applications in a secluded situation. Such tools include:

FileMon: This tool monitors file operations when the application is running on mobile device. It takes note of every executable file and gives a detail analysis.

RegMon: This tool is a registry monitoring utility which prompts users of applications that are accessing the registry. The accessed keys and registry data are all read in real-time.

Ethereal: This tool is a packet scanner that captures packets and supports the view contents on the device.

3. Performance evaluation and analysis of static and dynamic techniques

An evaluation of the performance of the various parameters is done with a far reaching correlation of their diverse traits. Table 1, gives the confinements of the static and dynamic approach of the malware identification techniques, while the malware discovery through both Static and Dynamic approach are given in Table 2 and Table 3 respectively.

Table 1: Limitations of Static and Dynamic Approaches

	Mechanism	Limitation
Static	Signature based detection	Cannot detect zero day malware and unknown malware types
	Permission based detection	May consider benign applications as malignant as a result of little contrasts between consents asked for by the two sorts.
	Dalvik byecode detection	It uses more memory
Dynamic	Taint Analysis	Not reasonable for real time examination
	Anomaly detection	Consume more battery and memory. It invoke more API calls.
	Emulation based detection	More resource consumption

Based on their working systems we have reasoned significant confinements and advantages for every recognition component.

Table 2. Malware detection by static analysis

Approach	Name	Goal	Method	Year	Limitations	Benefits

Signature Based Detection	DrordAnaly	Automatic collect; extraction, analysis and association of Android malwares	Create 3 level signature for app on the basis of API call; Perform Opcode level analysis (class, method, application)	13	Cannot detect unknown malware; similarity score may classify legitimate apps as malicious.	Also detect dynamic malware payloads; Associates malware at opcode level
	AndrodSimilar	Detect unseen and zero day samples of knowns malwares.	Use fuzzy hashing techniques; creates variable length signature and compares with signature database.	13	Limited signature database, can only detect knows malware	Effective against code obfuscation and repackaging.
Permission Based Detection	Stowaway	Application over privilege detection	API call tracing through static analysis tool; permission map to identify the permission required by each API call.	15	Cannot resolve complex reflective calls	Notify about the over privileged applications.
	R.Sato	Malware detection by manifest file analysis	Analyze manifest file; compare extracted information with keyword list; calculate malignancy score	15	Cannot detect adware samples	Light weight approach; low cost; can detect the unknown malwares
	PUMA	Malware detection	Analyze extracted permissions; evaluate the performance by k-fold cross with k=10	15	High false positive rate; not adequate for efficient malware detection	High detection rate
Dalvik Bytecode Detection	SCANDAL	Privacy leak detection	Extracts bytecode of applications as a dalvik executable file; translates dalvik executable language for efficient analysis.	13	More time and memory consumption; does not support application for privacy leakage	Saves data from privacy leakage; dalvik bytecode is always available; better accuracy
	DroidAPIMiner(42)	API level malware detection	Extract API level features; apply classifiers for evaluation	13	More occurrence of false positives; it generates incorrect classification	Better accuracy
	Karlsen	Dalvik bytecode formalization and control flow analysis	Provides formal control flow analysis; formalizes dalvik bytecode language with reflection features.	16	Require extension in analysis of reflection and concurrency handling	Support reflection and dynamic dispatch features; formal control flow analysis easily traces the API calls

4. Hybrid analysis detection technique

This is the blend of both static investigation and dynamic examination [24]. Android Application Sandbox is an example of hybrid technique proposed by [22] which distinguish suspicious applications by performing both static and dynamic investigation on them.

The procedure entails is that, it first checks for the presence of malware signature in the code under review and screens it. Thus, this strategy consolidates static and dynamic methods.

Table 3: Comparison of different dynamic analysis techniques

Approach	Name	Goal	Method	Year	Limitations	Benefits
Anomaly Detection	CrowDroid	Detecting anomalously behaving malicious application	Create tool to perform system calls tracing; dynamic analysis is performed on the data at server side; CrowDroid client app install on user's device	11	It required installation of CrowDroid client; results incorrect if legitimate app invokes more system calls	Provides deep analysis
	AntiMalDroid	Malware detection through characteristic learning and signature generation.	Generates behavioral characteristic; monitors the behavior of application and their characteristics; learning module	13	More time consumption	Low cost and better performance; higher detection rate
Taint Analysis	TaintDroid	Data stream investigation and spillage identification	Automatically leads the data; keeps track of the data	10	Cannot track information that leaves the device and return in network reply	Efficient tracking of sensitive information
Emulation based detection	AASandbox	Malware Detection	System calls tracking; built upon QEMU (quick emulator).	10	Limited code coverage	Can be utilized to enhance the effectiveness of the antimalware programs for Android OS
	DroidScope	Android malware detection	System calls tracking; built upon QEMU (quick emulator)	13	Limited code coverage	Can distinguish benefit acceleration assaults on the kernel

5. Conclusion

In this paper, malware and additionally their entrance strategies are evaluated while extensively reasoning their favorable circumstances and disservices. A proposition of a hybrid anti-malware is introduced to help address the impediments of the current static and dynamic strategies with the point of actualizing it sooner rather than later.

We have likewise studied on the different sorts of malware and classes of noxious programming. Specifically, the exposition of the different detection and instruments for mobile malware. In spite of the fact that the rate perils of new malware are expanding at a disturbing rate, there is careful investigation of instruments for dissecting malware with an unmistakable comprehension of different countermeasures that should be adopted.

It is determined that utilizing a static technique is less proficient at distinguishing the pernicious substance that are stacked progressively from remote servers. While the dynamic technique is proficient as it continues checking the application and ready to identify the vindictive substance at execution time, it is however obvious that, the segments of malicious codes that are not executed stay undetected. Clearly, any single security arrangement in cell phones cannot give full assurance against the vulnerabilities and malware. In this manner, it is smarter to convey more than one solution at the

same time - static and dynamic. Utilizing a crossover approach will first statically examine the application and will then perform a dynamic investigation. Despite the fact that the operation is costly because of accessibility of assets such as battery and memory, these restrictions of a hybrid usage can be tended to. Thinking about on twofold, right off the bat the static examination should be possible locally on the cell phone; and a short time later, the dynamic investigation could be performed in a conveyed design by sending the noxious action as a log record to a remote server. The remote server will play out the dynamic examination rapidly and productively as the server will have enough assets to perform a dynamic examination and can create quick reactions against the application behavior and notify the client. This hybrid solution needs more examination and is subjected to designs tradeoffs. Further studies will focus on how to make hybrid techniques more robust.

References

[1] B. Anderson, D. Quist, J. Neil, C. Storlie, and T. Lane, "Graph Based Malware Detection Using Dynamic Analysis," *Journal in Computer Virology,* vol. 7, pp. 247-258, 2011.

[2] B. Anderson, C. Storlie, and T. Lane, "Improving Malware Classification: Bridging the Static/Dynamic Gap," in Proceedings of 5th ACM Workshop on Security and Artificial Intelligen*ce, AISec*, pp. 3-14, 2012

[3] U. Bayer, P.M. Comparetti, C. Hlauschek, and C. Kruegel, "Scalable, Behavior-Based Malware Clustering.," in Proceedings of the 16th Annual Network and Distributed System Security Symposium*,* 2009.

[4] I. Firdausi, C. Lim, and A. Erwin, "Analysis of Machine Learning Techniques Used in Behavior Based Malware Detection," in Proceedings of 2nd International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT), Jakarta, 2-3 December 2010, pp. 201-203.

[5] D. Kong, and G. Yan, "Discriminant Malware Distance Learning on Structural Information for Automated Malware Classification.," in Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling, 2013.

[6] T. Lee, and J.J. Mody, "Behavioral Classification," in Proceedings of the European Institute for Computer Antivirus Research Conference (EICAR'06), 2006.

[7] R. Moskovitch, D. Stopel, C. Feher, N. Nissim, and Y. "Elovici, Unknown Malcode Detection via Text Categorization and the Imbalance Problem.," in Proceedings of the 6th IEEE International Conference on Intelligence and Security Informatics. 2008.

[8] S. Nari, and A. Ghorbani, "Automated Malware Classification Based on Network Behavior," in Proceedings of International Conference on Computing, Networking and Communications (ICNC), San Diego, 28-31 January 2013, pp. 642-647.

[9] J. Nieves, I. Santos, and P.G. Bringas, "Semi-Supervised Learning for Unknown Malware Detection," in," International Symposium on Distributed Computing and Artificial Intelligence Advances in Intelligent and Soft Computing, 2011.

[10] Norton. Norton Safe Web. of Computer Systems, July 2012.[Online]. Available: http://safeweb.norton.com

[11] I. Santos, J. Devesa, F. Brezo, J. Nieves, and P.G. Bringas, "OPEM: A Static-Dynamic Approach for Machine Learning Based Malware Detection," in Proceedings of International Conference CISIS'12-ICEUTE'12, Special Sessions Advances in Intelligent Systems and Computing, vol. 189, pp. 271-280, 2013.

[12] I. Santos, J. Nieves, and P.G. Bringas, "Collective Classification for Unknown Malware Detection.," in Proceedings of the International Conference on Security and Cryptography, Seville, 18-21 July 2011.

[13] G. Savan and B. Kaushal, "Techniques for Malware Analysis" in,' International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, Issue 4, April 2013ISSN: 2277 128X, April, 2013.

[14] R. Tian, L. Batten, and S. Versteeg, "Function Length as a Tool for Malware Classification," in Proceedings of the 3rd International Conference on Malicious and Unwanted Software, 2008.

[15] R. Tian, L. Batten, R. Islam, and S. Versteeg, "An Automated Classification System Based on the Strings of Trojan and Virus Families," in Proceedings of the 4th International Conference on Malicious and Unwanted Software, Montréal.

[16] R. Tian, M.R. Islam, L. Batten and S. Versteeg, "Differentiating Malware from Cleanwares Using Behavioral Analysis," in Proceedings of 5th International Conference on Malicious and Unwanted Software (Malware), Nancy, 2010, pp. 19-20 October 2010, 23-30, 2009.

[17] V. Mehra Dolly Uppal and V. Verma, Trend Micro. "A Brief History of Mobile Malware", *Basic survey on Malware Analysis, Tools and Techniques*, 2014.

[18] M.F. Zolkipli, and A. Jantan, "An Approach for Malware Behavior Identification and Classification," in Proceeding of 3rd International Conference on Computer Research and Development, Shanghai, pp.11-13, March 2011.

[19] Y. Zhou and X. Jiang, "Dissecting Android Malware: Characterizing and Evolution," in Proceedings of IEEE Symposium on Security and Privacy, August, 2012.

[20] N. DuPaul, "Common Malware Types," 12 October 2012. [Online]. Available: https://ww.veracode.com/blog/2012/10/common-malware-types-cybersecurity-101. [Accessed 21 December 2017]

[21] L.K. Yan and H. Yin, "Droidscope: Seamlessly Reconstructing the OS and Dalvik Semantic views for Dynamic Android Malware Analysis," in Proceedings of USENIX Security Symposium, 2012.

[22] T. Bläsing, L. Batyuk, A.D. Schmidt, S.A. Campete and S. Albayrak, "An android application sandbox system for suspicious software detection," in Proceedings of 5th IEEE International Conference on Malicious unwanted software, Malware, 2010.

[23] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer and Y. Weiss, "Andromal: a behavioural malware detection framework for android devices," in," Journal Intell. Inf. Syst., vol. 38, no.1, 2012.

[24] Y. Robiah, S. Rahayu, M.M. Zaki, S. Shahrin, M.A. Faizal and R. Marliza, "A new generic taxonomy on hybrid malware detection technique,"nin Journal arXiv preprint arXin: 0909.4860, 2009.

[25] B. Iker, Z. Urko and N.T. Simin, Crowdroid: behavior-based malware detection system for android," in Proceedings of the 1st ACM workshop on Security and privacy in smartphonesand mobile devices, 2011.

[26] "Mind the (Security) Gaps: The 1H 2015 Mobile Threat Landscape - Security News - Trend Micro USA." [Online]. Available: http://www.trendmicro.com/vinfo/us/security/news/mobile-safety/mindthe-security-gaps-1h-2015-mobile-threat-landscape. [Accessed: 08-Dec- 2017].