A COMPREHENSIVE REVIEW OF MOBILE BOTNET DETECTION USING GENETIC ALGORITHM: A SYSTEMATIC REVIEW

Muhammad Zuhair bin Abd Rahman, Madihah binti Mohd Saudi and Nurlida binti Basir
Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia, Nilai, Negeri Sembilan, Malaysia
E-mail: zuhairabdrahman@gmail.com

ABSTRACT

Nowadays, mobile botnet is considered as one of the biggest cyber threats attacking the smartphones especially on the Android platform. The loss of money, confidential information and productivity due to mobile botnet attacks to the smartphones, have triggered the formation of this research paper. This research paper presents a comprehensive review on the existing techniques in mobile botnet detection. A comparison with the existing works related with mobile detection techniques is further investigated and evaluated. Furthermore, this research paper explores the possibilities to integrate the genetic algorithm in mobile botnet detection to optimize the detection rate. Based on the comprehensive review made, it has been identified that the genetic algorithm is offering a promising result for a higher mobile.

Keywords: android, mobile botnet detection, genetic algorithm.

INTRODUCTION

United States Computer Emergency Response Team (US-CERT) has reported that malwares especially mobile botnet are currently having emerged as new threat targeting mobile platform (US-CERT, 2010). With the increasing number of smartphone users, mobile botnet has become as one of the major threats attacking mobile platform, especially the Android OS.

An infected device, known as bot can be part of a large connection of other infected devices, botnet. The botmaster behind this connection is in-charge and can control the infected devices (Feizollah *et al*., 2014). Once the devices got infected, these devices inform the botmaster and utilize it for spam, then initiate the distributed denial of service attack (DDoS), premium call and SMS syndicate and also stealing bank credential. Furthermore, statictics reported by F-Secure showed an increasing numbers of bots spread from January to March 2014 (F-Secure Labs, 2014).

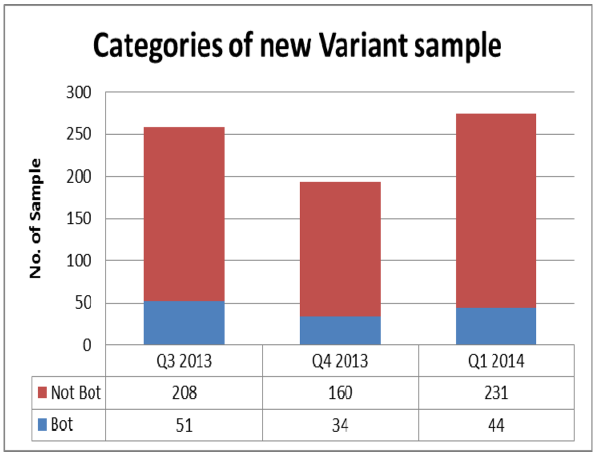


Figure-1. Categories of sample from new variant detected (F-Secure Lab, 2014).

ANDROID ARCHITECTURE

Anatomy of Android OS

Android was created by Android Inc. which was bought by Google and released as the Android Open Source Project (AOSP) in 2007. A group of 78 different companies formed the Open Handset Alliance (OHA) that is dedicated to develop and distribute Android. The software can be freely obtained from a central repository and modified in terms of the license which is mostly BSD and Apache (Google Inc., 2014).

Android was built from the ground-up, enable developers to create mobile applications that take full advantage of all a mobile device has to offer, open and now expanding towards tons of mobile devices. It is not just an operating system but a complete software stack that includes application framework, libraries and some core applications as shown in Figure-2. The Android architecture is made up of different components, which are composed into different layers (Maker and Chan, 2009).

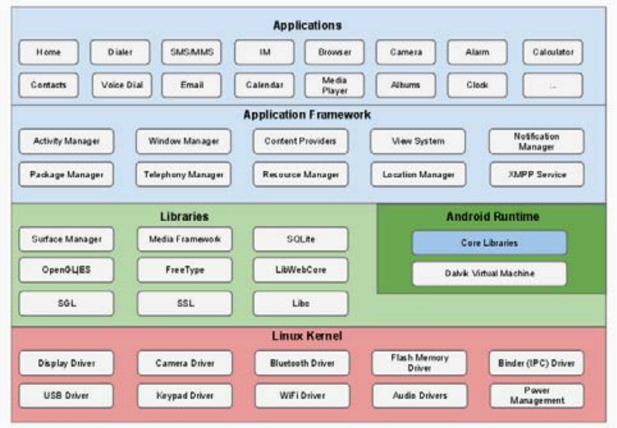


Figure-2. Android OS anatomy.

(Xianhua *et al*., 2009) and (Brahler, 2010) differentiate and describe the main component of Android Architecture as follows:

A. Applications

Any installed applications are positioned on top level in the framework, stretching from contact-app, SMS app, email client, to games and productivity-based like calculator and document editor. Application are created using the Java programming language.

B. Application framework

Developers have full access to the same framework APIs used by the core applications. It is also based on Java programming language. The application architecture is designed to simplify the reusing of all components. This mechanism allows every component to be replaced by the user.

C. Libraries

A set of C/C++ libraries is included and used by various components of the Android system. These capabilities are exposed to developers through the Android application framework.

D. Android runtime

The Android Runtime comprises of the Dalvik virtual machine and the Java core libraries. Every Android application runs in its own process given by the OS, and owns its own instance of the Dalvik virtual machine.

The Dalvik VM is executing files in .dex (Dalvik Executable) format in the application file which was optimized for minimal cpu-and-memory-usage. The Virtual Machine is register-based, and runs classes compiled by a Java language compiler that have been transformed at compile-time into the .dex format using the "dx" tool, that are shipped with the SDK.

E. Linux kernel

Android relies on Linux (the first Linux Kernel version 2.6, now version 3.1.4) for core system services such as memory management, process management, network stack, security, and driver model. The core also acts as a hardware abstraction layer between the applications and all the hardware.

MOBILE BOTNET DETECTION

Common botnet detection method

According to (Karim et al., 2014), botnet detection techniques are classified into two broad categories which are Honeynet and Intrusion Detection System (IDS).

i) Honeynet is one way of harvesting data and metadata from bots for deeper analysis. This includes analyzing the characteristic, the damage intensity of the attack, and also the technology used behind. Then, those information will be used to further investigation especially in locating the C&C server thus identifying the attacker and its motivation.

ii) IDS is a hardware-type or software-type application that inspects the incoming and outgoing network traffic for malicious activities or policy violations. IDS has the capability to respond to malicious traffic by blocking the users' IP from accessing the network. For IDS detection technique, it is classified as a signature based approach where the packets are compared with available database with signatures of known malicious threats and an anomaly based approach where it monitor and compare the traffic with an established baseline.

Note that this kind on detection usually applied on host machine such as computer and server and also on network-level while mobile botnet appear as a new kind of device that use low computational power, small screen and consume less bandwidth of data. Rather than using P2P and HTTP protocol, mobile botnet also capable to utilize other feature in smartphones such as SMS which will be discussed later.

Mobile botnet characteristic

(Pieterse and Olivier., 2012) had categorized the mobile botnet characteristic, which also related closely to botnet functionality that attack computers and servers such as communication with C&C server and also stealing credentials.

i. Repackaged Application: Legit application code modified by adding malicious and rehosted back on the application market. During installation, user will not notice the malicious intention of the application.

ii. Receiving commands: The most important characteristic is to be able to receive command from any remote server.

iii. Messaging: This can be used either for receiving command or to generate profit by exploiting the SMS app to subscribe with expensive premium SMS services. A lot of money can be generated here.

iv. Stealing Credentials: After receiving command, mobile botnet will grab any available information that may be valuable such as contact number, IMEI, GPS location, and the phone version.

v. Excessive permission request: In the application, it is required to state what kind of service or hardware it will request to access. Usually in repackaged application, it will prompt to request tons of features required by the malicious code to run hiddenly in smartphone.

vi. Additional Content Download: This ability usually triggered by disguised app which is legit during installation until some time, it trigger to download update so it will enhance the malicious ability.

Techniques used in detecting mobile botnet

In previous years, many studies have been conducted on mobile botnets such as by (Traynor et al., 2009), (Mulliner and Seifert, 2010), (Zeng et al., 2010), (Guerard and Park, 2013), (Zhou and Jiang, 2012). Traynor *et al.* studied the possibility of using bluetooth as the C&C channel of a botnet while (Mulliner and Seifert, 2010) proposed SMS-HTTP command and control system in which the attacker created command and then the command was sent to bots via SMS. The command is then being uploaded to a designated website in an encrypted file. Then, each bot will download, decrypt the file, and send out the commands to other bots via SMS. (Zeng et al., 2010) designed a SMS-P2P hybrid botnet which uses SMS as the C&C channel, and the peer-to-peer (P2P) network as the underlying structure. Botnet communicates by obtaining commands in a P2P fashion by sending and receiving SMS messages.

Guerard *et al.* have created a customized piece of malware to investigate the capabilities and limitations of Android malware authors and anti-virus detection (Guerard and Park, 2013). The sample in which equipped with *zero-day attack*, capable to penetrate the Android market without being detected by signature based anti-virus. (Zhou and Jiang, 2012) also mentioned the incapability of the Android anti-virus software to protect against attacks of known samples, where it was only able to capture 20% of the malwares. The capability of the sample that has been released 2 years ago, to bypass the anti-virus detection is a challenge for researchers and the anti-virus company to solve it immediately.

Feature used in mobile botnet detection technique.

Prior doing the mobile botnet classification, the mobile botnet feature must be extracted from the sample. The Android application is packaged as .apk file (Maker and Chan, 2009). The summarization of the existing works on feature extraction for mobile detection can be referred in Table-1.

Table-1. Summary on feature extracted for detection.

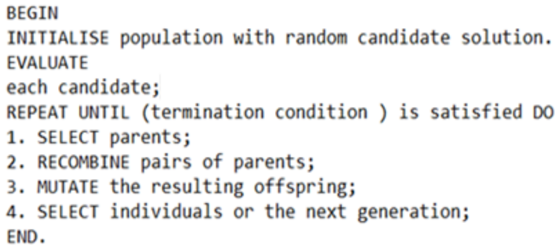
Paper Title	Feature Used	Strength	Weaknesses
Crowdroid (Burguera *et al*., 2011)	Android System Call	Good detection as Genuine app issue different type and number of System Call compared to malicious app	False-positive more likely to occur if the apps make use of less system call
MADAM: Mu-lti-level Anomaly Detector for Android Malware (Hanumantha *et al*., 2011)	13 feature based on user and kernel level	The overall detection accuracy was of 100% for all malware	MADAM is able to detect an intrusion attempt but it is not able to detect the malicious source.
PUMA: Permission usage to detect Malware in Android (Sanz *et al*., 2013)	Permission	High Detection Rate	High False-Positive rate
Performance Evaluation Permission-Based Detection for Android Malware (Chun-Ying, H. *et al*., 2013)	Permission	Permission is the first layer of defense on android	The performance of detection is not very good, only 81%
An Android Application Sandbox System For Suspicious Software Detection (Bläsing *et al*., 2013)	System call, library calls	Measurements are very diverse, delivering a very high entropy dataset	Yet to be found, only done POC
Detecting Android Malware by Analyzing Manifest files (Sato *et al*., 2013)	Permission, Intent filter (action, category) Process name	Only use manifest file, 90% rate of detection	The sample is old, retrieved before September 2011
A Machine Learning Approach to Android Malware Detection (Sahs *et al*., 2012)	Permission, Control Flow Graph	Low false-negative rate	High false-positive rate
A Study of Machine Learning Classifier For Anomaly-Based Mobile Botnet Detection (Feizollah *et al*., 2013)	(Network) TCP_size, Connection Duration, No. of parameter in GET/POST request	Can achieve up to 99.94% detection rate	No guarantee on new sample, system need more new training dataset
Detection of Malicious Android Mobile Applications Based on Aggregated System Call Events (Joung Ham *et al*., 2014)	System Call	Improved version of *Crowdroid*, better malware detection	The existing weakness in Crowdroid, is not addressed, no improvement on similar issue.

Therefore, based on the weaknesses identified in Table-1, it is very significant and a need to have a standard operating procedure (SOP) in extracting the feature in the dataset. A proper SOP in feature extraction helps to increase the detection rate.

GENETIC ALGORITHM

Structure of genetic algorithm

Genetic algorithm (GA) was developed by John Holland and his team, is a subset of Evolutionary algorithm where it was inspired by nature's reproductive system, in which the population is represented as genotype, decoded and evaluated for fitness. From there, only the fittest individuals in a generation will undergo optimisation process such as selection, recombination and mutation to produce subsequent generations (Goldberg, 1989; Holland, 1992). Figure-3 shows the model on how the Genetic Algorithm works.



```
BEGIN
INITIALISE population with random candidate solution.
EVALUATE
each candidate;
REPEAT UNTIL (termination condition ) is satisfied DO
1. SELECT parents;
2. RECOMBINE pairs of parents;
3. MUTATE the resulting offspring;
4. SELECT individuals or the next generation;
END.
```

Figure-3. Early genetic algorithm model (Holland, 1992).

Botnet detection with genetic algorithm implementation

Genetic Algorithm (GA) has evolved as several studies had been published on malware detection integrated with GA such as by (Noreen *et al*., 2009; M.N. Yusoff *et al*., 2011; Aldwiri and Alsalman, 2012; Edge *et al*., 2006; Mathew *et al*., 2014).

(Noreen *et al*., 2009) had done a proof-of-concept study where they evolve a well-known virus family, called Bagle into new viruses, previously unknown and knownvariants of Bagle using Genetic Algorithm. Out of a population of 2000 sample, only 50% of it is matched with the original Bagle sample while others are new variant of it. Testing with commercial anti-virus solution shows not all populations are detected, conclude that software can be evolved, inspire softwares, especially malware detector to ensure reusability rather than redeveloped from scratch.

(M.N. Yusoff *et al*., 2011) implemented GA to improve classification and the accuracy rate of portable executable (PE) file that failed to be classified by decision tree classifier while (Aldwiri and Alsalman, 2012) also implemented GA to expand the training dataset through mutations. It managed to reduce training time and increased the detection from 77% to 80%. While (Edge *et al*., 2006) had applied GA in REALGO AIS algorithm by training and modifying the antibodies where if a program match the antibody beyond a specified threshold, then the antibody is updated so the detection of malicious code improved. Despite the fact that REALGO is an AIS-based algorithm, GA actually improve the algorithm to find better solution. (Mathew *et al*., 2014) implemented (GA) into the layered system to detect and filter http botnet attack and has succeeded provide less false positive rate.

Most of the existing works discussed above were not focusing on mobile botnet and targeted for Windows platform only. Therefore, with the existing theory and model especially by (Noreen *et al*., 2009), it is possible to integrate Genetic Algorithm with mobile botnet detection by evolving the mobile botnet dataset sample and train the malware detector to continue improve the detection rate to achieve better and more higher detection rate. Other capability includes predicting the future variant of mobile botnet based on the variant evolved by using GA also can be further explored.

Table-2: Summary on botnet detection integrated with genetic algorithm.

Proposed method	Rationale	Strength
Noreen *et al*., 2009	Evolve *Bagle* virus into new set of unknown and known- variants of viruses using Genetic Algorithm	Creation of more dataset and high-level feature representation of *Bagle* virus, mostly not yet detected by commercial antivirus.
M.N. Yusoff *et al*., 2011	Using GA to improve classification and the accuracy rate of portable executable(PE) file that failed to be classified by decision tree classifier	Results show that accuracy rate increases as the value of threshold increases.
Aldwiri and Alsalman, 2012	Expanding dataset using GA to reduce training time.	Detection improved from 77% to 80%
Edge *et al*., 2006	Using GA to train antibodies so it would be up-to-date and improve detection.	The antibodies are tuned to differentiate between good and bad processes and programs.
Mathew *et al*., 2014).	Implement GA into layered system to detect and filter HTTP botnet	Manage to produce less false positive rate result.

CONCLUSIONS

The number of mobile devices that use Android OS is increasing tremendously all over the world and it is expected to be more with the introduction of Internet of Things(IoT) that enable more mobile computing devices to be implemented anywhere (Malaysia Communication and Multimedia Commission, 2013). A new and better intelligent malwares and variants will be created to exploit the smartphones for malicious intention. It is a challenge for the researchers all over the world to develop a method or an algorithm to detect such new intelligent threat efficiently. For future work, a new model using Genetic Algorithm to improve the mobile botnet detection will be developed by the researchers of this research paper.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Universiti Sains Islam Malaysia (USIM) dan Institut Sains Islam (ISI), USIM for the support and facilities provided. This research paper is supported by Ministry of Education Malaysia under FRGS grant [FRGS/1/2014/ICT04/USIM/02/1], RAGS grant [USIM/RAGS/FST/36/51013] and also Universiti Sains Islam Malaysia grant [PPP/FST/SKTS/30/12712].

REFERENCES

Adrien Guerard and Danny Park (2013). Analyzing Android Malware and Current Detection Systems, Swarthmore College Computer Science Senior Conference (CPSC-97), Autumn, pp. 159-169.

Aldwairi, M., and Alsalman, R., (2012). MALURLS: A Lightweight Malicious Website Classification Based on URL Features, Journal of Emerging Technologies in Web Intelligence, Vol4, No 2.

Ali Feizollah ,Nor Badrul Anuar,Rosli Salleh (2014). A Study Of Machine Learning Classifiers for Anomaly-Based Mobile Botnet Detection, Malaysian Journal of Computer Science, Volume 26, Issue 4, pp 251-265.

Brady, P. (2009). Android Anatomy and Physiology, Google I/O Developer Conference.

Brahler, S. (2010). Analysis of the android architecture. Karlsruhe institute for technology.

Bläsing, T.; Batyuk, L.; Schmidt, A-D.; Camtepe, S.A; Albayrak, S. (2010). An Android Application Sandbox system for suspicious software detection, Malicious and Unwanted Software (MALWARE), 5th International Conference on , pp.55,62, 19-20 Oct. 2010, doi: 10.1109/MALWARE.2010.5665792.

Brady, P., (2008). Android Anatomy and Physiology, Google I/O Developer Conference.

Breedam, A. (2001). Comparing descent heuristics and metaheuristics for the vehicle routing problem. Computers & Operations Research, 28(4), pp.289--315.

Burguera, I., Zurutuza, U., and Nadjm-Tehrani, S., (2011). Crowdroid: behavior-based malware detection system for Android. In Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices (SPSM '11). ACM, New York, NY, USA, 15-26. DOI=10.1145/2046614.2046619 http://doi.acm.org/10.1145/2046614.2046619.

Chun-Ying H., Yi-Ting, T., Chung-Han H., (2013). Performance Evaluation on Permission-Based Detection for Android Malware, Advances in Intelligent Systems and Applications - Volume 2, Smart Innovation, Systems and Technologies Volume 21, pp 111-120.

D. Goldberg (1989). Genetic Algorithm in search, Optimization and Machine Learning, Addison-Wesley, Reading, MA.

Dini, G., Martinelli, F., Saracino, A., and Sgandurra, D., (2012) MADAM: a multi-level anomaly detector for android malware. In Proceedings of the 6th international conference on Mathematical Methods, Models and Architectures for Computer Network Security: computer network security (MMM-ACNS'12), pp 240-253. DOI=10.1007/978-3-642-33704-8_21 http://dx.doi.org/10.1007/978-3-642-33704-8_21

F-Secure Labs (2014). Mobile Threat Report Q1 2014. F-Secure Corporation. http://www.f-secure.com/documents/996508/1030743/Mobile_Threat_Report_Q1_2014_print.pdf. (Accessed on 20 Oct 2014)

Farahani, S., Abshouri, A., Nasiri, B. and Meybodi, M. (2011). A Gaussian firefly algorithm. International Journal of Machine Learning and Computing, 1(5), pp.448-453.

Feizollah, A., Nor Badrul Anuar, Salleh R., (2014). A Study Of Machine Learning Classifiers for Anomaly-Based Mobile Botnet Detection, Malaysian Journal of Computer Science, Volume 26, Issue 4, Pg251-265.

Gandomi, A., Yang, X. and Alavi, A. (2011). Mixed variable structural optimization using firefly algorithm. Computers & Structures, 89(23), pp.2325-2336.

Google Inc. (2014). Android Open Source Project (AOSP), http://source.android.com/, Accessed on 30 April.

J. Holland (1992). Adaptation in Natural and Artificial System, 2nd, MIT Press, Cambridge, MA.

Joung Ham, Y., and Hyung-Woo Lee, (2014). Detection of Malicious Android Mobile Applications Based on Aggregated System Call Events, IJCCE, Vol.3(2): 149-154 ISSN: 2010-3743, DOI: 10.7763/IJCCE.2014.V3.310.

K.S. Edge, Lamont, G.B., and Raines, R.A., (2006). A Retrovirus Inspired Algorithm for Virus Detection & Optimization. In Proceedings of the 8th annual conference on Genetic and evolutionary computation (GECCO '06). ACM, New York, NY, USA, pp103-110. DOI=10.1145/1143997.1144016 http://doi.acm.org/10.1145/1143997.1144016

Karim, A., Rosli Bin Salleh, Shiraz, M., Syed Adeel Ali Shah, Awan, I., Anuar, N.B., (2014). Botnet Detection Techniques: Review, Future Trends and Issues Accepted for Publication in Journal of Zhejiang University-SCIENCE C-Computers & Electronics, February 2014.

M.N. Yusoff and Jantan, A., (2011). Optimizing Decision Tree in Malware Classification System by using Genetic Algorithm, International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(3): pp 694-713.

Maker, F., Chan, Y., (2009). A Survey on Android vs. Linux, University of California.

Malaysia Communication and Multimedia Commission, (2013). Statistical Brief Number Fourteen Hand Phone Users Survey 2012, ISSN 1823-2523.

Mathew, S.E., Ali, A., Stephen, J., (2014). Genetic Algorithm based Layered Detection and Defense of HTTP Botnet, ACEEE International Journal on Network Security, Vol. 5, No. 1, January 2014.

Mulliner, C. and Seifert, J., P.(2010). Rise of the iBots: 0wning a telco network. Proc. MALWARE, France.

Noreen, S., Murtaza, S., Shafiq, M.Z., and Farooq, M., (2009). Evolvable malware, In Proceedings of the 11th Annual conference on Genetic and evolutionary computation (GECCO '09). ACM, New York, NY, USA, 1569-1576. DOI=10.1145/1569901.1570111 http://doi.acm.org/10.1145/1569901.1570111.

Pieterse, H.; Olivier, M.S. (2012). Android botnets on the rise: Trends and characteristics. Information Security for South Africa (ISSA), 2012 , vol., pp.1,5, 15-17 Aug. 2012, doi: 10.1109/ISSA.2012.6320432.

Sahs, J.; Khan, L. (2012). A Machine Learning Approach to Android Malware Detection, Intelligence and Security Informatics Conference (EISIC), European, pp.141,147, 22-24 Aug. 2012, doi: 10.1109/EISIC.2012.34.

Sanz, B., Santos, I., Laorden, C., Xabier Ugarte-Pedrero, Pablo Garcia Bringas, Álvarez, G., (2013). PUMA: Permission Usage to Detect Malware in Android, International Joint Conference CISIS'12-ICEUTE´12-SOCO´12 Special Sessions, Advances in Intelligent Systems and Computing Volume 189, pp. 289-298.

Sato, R., Chiba, D., Goto, S. (2013). Detecting Android Malware by Analyzing Manifest Files, Proceedings of the Asia-Pacific Advanced Network, Volume 36, pp. 23-31.

Traynor, P., Lin, M. and Ongtang, M. (2009). On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core. In: Proceeding CCS, Chicago, USA.

US-CERT (2010), Technical Information Paper-TIP-10-105-01, Cyber Threats to Mobile Devices, accessed from https://www.us-cert.gov/security-publications on 4/3/2014.

Xianhua Shu, Zhenjun Du, and Rong Chen. (2009). Research on mobile location service design based on android. In Proceedings of the 5th International Conference on Wireless communications, networking and mobile computing (WiCOM'09). IEEE Press, Piscataway, NJ, USA, pp. 5166-5169.

Zeng, Y., Hu, X. Shin, K., G. (2010). Design of SMS Commanded-and-Controlled and P2P-Structured Mobile Botnets. University of Michigan Technical Report.

Zhou, Y., & Jiang, X. (2012). Dissecting Android Malware: Characterization and Evolution. 2012 IEEE Symposium on Security and Privacy, (4), 95–109. doi:10.1109/SP.2012.16.