MJoSHT
Malaysian Journal of Science, Health & Technology
[mjosht.usim.edu.my]

USIM

Article

A Systematic Review Analysis for Mobile Botnet Detection using GPS Exploitation

Muhammad Yusof [1,a], Madihah Mohd Saudi[1,b, 2], Farida Hazwani Mohd Ridzuan[1,c, 2]

[1]*Faculty of Science and Technology (FST)*

E-mail: [a]*muhammad.yusof@raudah.usim.edu.my*, [b]madihah@usim.edu.my, [c]farida@usim.edu.my

[2]*Cybersecurity and System Research Unit, Islamic Science Institute (ISI),*
Universiti Sains Islam Malaysia, Nilai, Negeri Sembilan, Malaysia

***Abstract*— At present, mobile botnet has become a cyber threat for smartphone users especially on the Android platform. It has the capabilities to exploit the vulnerabilities and steal confidential information in the victim's smartphone. Zeus, DroidDream and MisoSMS are examples of mobile botnets that have affected thousands of users worldwide. Therefore, this research paper presents a systematic review analysis on the existing techniques for mobile botnet detection techniques. It discusses the strengths and the weaknesses of the existing mobile botnet detection techniques and related works of mobile botnet that exploit GPS. This research paper can be used as a reference and guidance for those with the same interest.**

***Keywords*— Android; Mobile Botnet Detection; GPS Exploitation.**

I. INTRODUCTION

The popularity of mobile devices becomes the major threat to the users. Current usage of mobile is not restricted only for making calls or sending messages, but open for different online services such as online banking, social networking and also for web browsing. 13% market grew for smartphones in Q2 2015 compared to the previous year with shipments of 341.5 million as per the record of International Data Corporation (IDC) [1]. Meanwhile, among all mobile operating systems (e.g. Blackberry OS, Windows, iOS), Android is the most popular and dominant one in the market with 82.8% market share as reported by IDC (2015) for Q2 2015. The Open Handset Alliance along with Google had developed an OS based on the Linux kernel for mobile devices, such as smartphones and tablets and named it Android [2].

The main issue with these handheld devices are that they are targeted by malware, especially by the mobile botnet. Mobile botnets infect mobile devices with a particular virus or malware without the knowledge or consent of users and give the attacker the ability to remotely control them [3]. The command and control (C&C) mechanism is used to communicate with these infected devices. The attacker uses botmaster to control it and can commit cyber-attacks or cyber-crimes; e.g. gathering of sensitive information to exploit the user or use them for illegal purposes; interrupting with denial of services (DoS); sending spam messages.

This paper is structured in sections. A comprehensive review of mobile malware detection methods is presented in section 2 and section 3 highlights the previous study on mobile malware that was specifically involved with GPS exploitation. At the end, Section 4 concludes this paper.

II. MOBILE BOTNET DETECTION

Numerous studies and surveys have discussed about mobile botnets detection in general. Felt et al. [4] was the first who looked into mobile malware and described all kinds of mobile operating systems including the Symbian malware from 2009 to 2011. They also summarized all types of mobile malware, but the research dataset was too small. Meanwhile, a study was conducted by La Polla et al. [3] where they conferred security on threats and vulnerabilities, discussed about the security solutions pertaining to mobile devices from 2004 till 2011 and discussed general issues about Android.

Eslahi et al. [5] studied mobile botnet in general, including mobile botnet command and control mechanisms, malicious activities, current challenges, and issues in mobile botnet detection. While Nigam [6], presented a comparison between mobile and PC botnets, their fundamentals, as well as conceptual and implementation differences. This research summarized and analysed all the known mobile botnets, including their variants highlighting their differences and commonalities. The most comprehensive survey concerning feature selection on mobile malware detection was carried out by Feizollah et al. [7] by studying 100 previous research

works published between 2010 and 2014. On the other hand, a taxonomy review by Karim et al. [8] depicted the mobile botnet attacks by exploiting detection approaches, operational impact, vulnerabilities, target audience, platform, and mobile botnet architecture. While Rahman et al. [9], developed a system implementing a genetic algorithm for mobile botnet detection. They refined the research by reviewing other research works that employed bio-inspired or evolutionary algorithm for mobile botnet detection.

In recent years, many researches were conducted on mobile botnets detection. In this paper, all the research works from 2011 to 2016 are tabulated in Table 1 mentioning the method, selected features their strengths and weaknesses of the work are also summarised.

TABLE I

SUMMARY OF EXISTING RESEARCH ON MOBILE MALWARE DETECTION

Authors	Method of Detection	Feature Used	Strength	Weakness
Choi et al. [10]	Inspection of "pull" style C&C traffic's flow features moving through VPN. Abnormal models, whitelist and signatures were used in the developed detection system.	Network traffic	Use of only abnormal models provided higher detection rate of 94.6%. Addition of signatures and whitelist offered FP rate of 0%.	The detection rate was compared with PC botnet.
Shabtai et al. [11]	Continuous monitoring of different events and features that were obtained from the mobile device was done by the host-based malware detection system framework and the classification of the collected data was done by applying Machine Learning anomaly detectors	88 features	Light-weight application, installed in the mobile device and consumption of lower power.	Not using a real malware for the testing.
Dini et al. [12]	Anomaly-based IDS using machine learning techniques.	13 features based on user and kernel level	Obtained accuracy rate was 93% for 10 malware.	Incapable of detecting malware that avoids the system call with root permission
Sahs and Khan [13]	One-Class Support Vector Machine for malware detection and Control Flow Graph (CFG) for input application.	Permissions	Very low false negative rate.	To test the Java code, metadata and similar features.
Grace et al. [14]	Filtering of applications from zero-day malware or Android Google Play market.	Permissions	High, medium and low – these three risk categories were introduced by the prototype.	Detection scheme depends on signatures only and maybe miss encrypted or obfuscated exploits
Yerima et al. [15]	Using data mining Bayesian Classification	Permissions	Improved detection rates in comparison to the common signature-based antivirus software using the similar sample.	1000 samples from 49 Android malware families and 1000 benign applications are not enough for the sampling
Aung and Zaw [16]	Machine learning-based with K-means clustering	Permissions	Highly positive rate	Testing of 500 samples of android apps, not enough data sample
Burguera et al. [17]	Based on crowdsourcing, traces were collected from an unlimited number of real users by embedding detector in an overall framework.	System call	This framework use of a crowdsourcing system to collect data from users and analyzing this data in	Less system call by the apps increases False-positive rate more likely.

Authors	Method of Detection	Feature Used	Strength	Weakness
			the remote server.	
Arp et al. [18]	Identification of malicious applications directly enabled in the Android smartphone by lightweight method.	Permissions, intent filter, network address, hardware component	Using many statics features. Detection rate 94% with low false alarm for malware	Limit to static analysis only.
Sanz et al. [19]	Detection of malware by extracting features from the Manifest file of the applications and classification through machine learning.	Permissions	The evaluation and comparison between machine learning on mobile malware detection	The detection ratio can be improved by using other features of the applications.
Sanz et al. [20]	The permissions were extracted from the application itself and analyzed by machine learning to detect malicious Android applications.	Permissions	High detection rate	High false positive rate. Using dynamic analysis could improve malware detection.
Wu et al. [21]	Malware detection system was developed by using API calls along with the manifest files by utilising various machine learning algorithms like Naive Bayes, k-nearest neighbors and k-means.	Permissions, intent filters and API calls	Accuracy up to 97.87%. This approach is better than Androguard.	Cannot detect Android malware with a single sample
Karim et al. [22]	Automatic mobile botnet detection implementing machine learning methods. The framework consisted of 3 components, namely dynamic analysis, feature mining and learning.	Network traffic	Accuracy 99.49%	The sample file size limits to 8MB
Sato et al. [23]	Only the required manifest files of Android applications were analyzed by a lightweight method.	Permission and intent filter (Manifest file)	Accuracy 90% with only manifest file	Old samples that were retrieved before September 2011 were used.
Feizollah et al. [24]	Mobile botnet detection methods using five machine learning classifiers such as support vector machine, multi-layer perceptron, decision tree, k-nearest neighbour, and Naïve Bayes were compared.	Networks parameters such connection duration, TCP size and number of GET/POST.	k-nearest neighbour classifier achieved TP as high as 99.94% and FP by 0.06%	Comparison limit to 5 types of machine learning classifiers.
Isohara et al. [25]	Kernel-based behavior analysis for Android malware inspection which consists of a log collector in the Linux layer and a log analysis application.	System calls	The unknown application's malicious behavior was effectively detected as per the obtained results.	The samples too small.
Kang et al. [26]	Static analysis was done by using creator information as a feature in Android malware detection method and it classified malicious applications into similar groups for better performance.	Behaviors and permissions	The accuracy of detection and classification were 98% and 90% respectively.	Should be applied both with dynamics analysis features.
Rastogi et al. [27]	AppsPlayground for Android is a framework for automatic analysis of smartphone applications.	API calls, system calls and Java code	Malicious functionality and privacy leaks of the applications were detected automatically and	Taint analysis taking longer time.

Authors	Method of Detection	Feature Used	Strength	Weakness
			effectively.	
Seo et al. [28]	Identification of the presence of root exploits and potential vulnerabilities of Android apps.	API calls and permissions	Useful in detecting malicious mobile apps for home security systems, airplane tracking and booking systems, and online banking systems.	Both the static and dynamic analysis can be combined and used in obfuscation technique to detect effectively.
Wu et al. [29]	A machine learning method controlled the use of data flow in application program interfaces (APIs) as classification features in detecting Android malware.	API calls	Unknown Android malware was detected with 97.60% accuracy. Nearly 40% reduction in the time overhead of static privacy leakage analysis.	Limited to static analysis.
Shabtai et al. [30]	Meaningful deviations in a mobile application's network were identified using behavior-based anomaly detection system.	Network application-level	Protection of mobile device users from malicious attacks on their phones.	Overhead of the Features Extractor process was not measured.
Suarez-Tangil et al. [31]	Malware detection system using text mining and retrieval techniques of information.	Code analysis	The developed system was accurate, scalable and fast as per the experimental results.	Obfuscation code can defeat this classification

III. GPS EXPLOITATION

Not many researchers had focused on GPS exploitation in Android mobile OS. Ma et al. [32] investigated the location information leakage in Android and proposed a new tool namely, Brox to identify a potential information leakage path in android malicious applications. While Vanjire et al. [33] developed an Android Application based on location-based system (LBS) that offered varied location-based services, such as changing of mobile profile from normal to silent mode and vice versa, for certain places where the users had registered. The LBS function in Android or any mobile devices can also find out the locations of nearest friends and family members. However, no elaboration was offered on how the cloud for the anti-malware provider could detect the leakage of information in Android application, as well as privacy issues, on GPS and LBS.

Other than that, Singhal and Shukla [34] implemented the LBS via Google Web Services and Walk Score Transit APIs on Android to offer several services to the users based on their location. This paper, nonetheless, just highlighted the implementation of LBS. Note that the discussion on the security of this implementation is not provided.

IV. CONCLUSION

This paper reviewed the earlier research studies on mobile botnet detection. All of them proposed various detection methods to combat the mobile botnets. In fact, based on the current report prepared by McAfee Labs [35], the number of mobile malware that have been found will continue to increase day by day in parallel with the emerging smartphones technologies. Hence, it can be concluded that the recent mobile botnet detection methods still need enhancement to achieve the desired accuracy. Therefore, there is space for further research on mobile botnet classification and detection mechanism and they can be proved significant.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Universiti Sains Islam Malaysia (USIM) for the support and facilities provided. This research paper is supported by the Ministry of Higher Education (MOHE), Malaysia grant: [USIM/FRGS/FST/32/50114] and [PPP/UCG-0114/FQS/30/11714].

REFERENCES

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

[2] Google Inc., "Android Open Source Project (AOSP)," 2016. [Online]. Available: http://source.android.com/. [Accessed: 12-Oct-2016].

[3] M. La Polla, F. Martinelli, and D. Sgandurra, "A Survey on Security for Mobile Devices," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 1, pp. 446–471, 2012.

[4] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, "A survey of mobile malware in the wild," in *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices - SPSM '11*, 2011, pp. 3–14.

[5] M. Eslahi, M. R. Rostami, H. Hashim, N. M. Tahir, and M. V. Naseri, "A Data Collection Approach for Mobile Botnet Analysis and Detection," *2014 IEEE Symp. Wirel. Technol. Appl.*, pp. 199–204,

2014.
[6] R. Nigam, “A Timeline Of Mobile Botnets,” *Virus Bulletin*, vol. Spring, no. March, pp. 1–8, 2015.
[7] A. Feizollah, N. B. Anuar, R. Salleh, and A. W. A Wahab, “A review on feature selection in mobile malware detection,” *Digit. Investig.*, vol. 13, pp. 22–37, 2015.
[8] A. Karim, S. A. A. Shah, R. Bin Salleh, M. Arif, R. Md Noor, S. Shamshirband, S. Adeel, A. Shah, R. Bin Salleh, M. Arif, and R. Noor, “Mobile botnet attacks - an emerging threat: classification, review and open issues,” *KSII Trans. Internet Inf. Syst.*, vol. 9, no. 4, pp. 1471–1492, 2015.
[9] M. Z. A. Rahman, M. M. Saudi, and N. Basir, “A Comprehensive Review of Mobile Botnet Detection Using Genetic Algorithm : A Systematic Review,” *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 3, pp. 1399–1404, 2015.
[10] B. Choi, S.-K. Choi, and K. Cho, “Detection of Mobile Botnet Using VPN,” in *Proceeding of the 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2013, pp. 142–148.
[11] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, “‘Andromaly’: a behavioral malware detection framework for android devices,” *J. Intell. Inf. Syst.*, vol. 38, no. 1, pp. 161–190, 2012.
[12] G. Dini, F. Martinelli, A. Saracino, and D. Sgandurra, “MADAM: a Multi-Level Anomaly Detector for Android Malware,” in *Proceeding of the International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security. Lecture Notes in Computer Science.*, 2012, pp. 240–253.
[13] J. Sahs and L. Khan, “A Machine Learning Approach to Android Malware Detection,” in *Proceeding of the 2012 European Intelligence and Security Informatics Conference*, 2012, pp. 141–147.
[14] M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, “RiskRanker : Scalable and Accurate Zero-day Android Malware Detection Categories and Subject Descriptors,” in *Proceedings of the 10th international conference on Mobile systems, applications, and services, MobiSys ’12*, 2012, pp. 281–293.
[15] S. Y. Yerima, S. Sezer, G. McWilliams, and I. Muttik, “A New Android Malware Detection Approach Using Bayesian Classification,” in *Proceeding of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, 2013, pp. 121–128.
[16] Z. Aung and W. Zaw, “Permission-Based Android Malware Detection,” *Int. J. Sci. Technol. Res.*, vol. 2, no. 3, pp. 228–234, 2013.
[17] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, “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*, 2011, p. 15.
[18] D. Arp, M. Spreitzenbarth, H. Malte, H. Gascon, and K. Rieck, “Drebin: Effective and Explainable Detection of Android Malware in Your Pocket,” in *Proceeding of the Symposium on Network and Distributed System Security (NDSS)*, 2014, pp. 23–26.
[19] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, J. Nieves, P. G. Bringas, and G. Álvarez Marañón, “Mama: Manifest Analysis for Malware Detection in Android,” *Cybern. Syst. - Intell. Netw. Secur. Surviv.*, vol. 44, no. 6–7, pp. 469–488, 2013.
[20] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, P. G. Bringas, and G. Alvarez, “PUMA: Permission Usage to detect Malware in Android,” in *Proceeding of the International Joint Conference CISIS’12-ICEUTE´12-SOCO´12 Special Sessions*, 2013, pp. 289–298.
[21] D.-J. Wu, C.-H. Mao, T.-E. Wei, H.-M. Lee, and K.-P. Wu, “DroidMat: Android malware detection through manifest and API calls tracing,” in *Proceedings of the 2012 Seventh Asia Joint Conference on Information Security*, 2012, pp. 62–69.
[22] A. Karim, R. Salleh, and M. K. Khan, “SMARTbot: A Behavioral Analysis Framework Augmented with Machine Learning to Identify Mobile Botnet Applications,” *PLoS One*, vol. 11, no. 3, p. e0150077, 2016.
[23] R. Sato, D. Chiba, and S. Goto, “Detecting Android Malware by Analyzing Manifest Files,” in *Proceedings of the Asia-Pacific Advanced Network*, 2013, vol. 36, pp. 23–31.
[24] A. Feizollah, N. B. Anuar, R. Salleh, F. Amalina, R. R. Ma’arof, and S. Shamshirband, “A study of machine learning classifiers for anomaly-based mobile botnet detection,” *Malaysian J. Comput. Sci.*, vol. 26, no. 4, pp. 251–265, 2013.
[25] T. Isohara, K. Takemori, and A. Kubota, “Kernel-based behavior analysis for android malware detection,” in *Proceedings of the 2011 Seventh International Conference on Computational Intelligence and Security*, 2011, pp. 1011–1015.
[26] H. Kang, J. W. Jang, A. Mohaisen, and H. K. Kim, “Detecting and Classifying Android Malware Using Static Analysis along with Creator Information,” *Int. J. Distrib. Sens. Networks*, vol. 2015, p. 9, 2015.
[27] V. Rastogi, Y. Chen, and W. Enck, “AppsPlayground : Automatic Security Analysis of Smartphone Applications,” in *In Proceeding of the 3rd ACM conference on Data and Application Security and Privacy)*, 2013, pp. 209–220.
[28] S. H. Seo, A. Gupta, A. M. Sallam, E. Bertino, and K. Yim, “Detecting mobile malware threats to homeland security through static analysis,” *J. Netw. Comput. Appl.*, vol. 38, no. 1, pp. 43–53, 2013.
[29] S. Wu, P. Wang, X. Li, and Y. Zhang, “Effective Detection of Android Malware Based on the Usage of Data Flow APIs and Machine Learning,” *Inf. Softw. Technol.*, vol. 75, pp. 17–25, 2016.
[30] A. Shabtai, L. Tenenboim-Chekina, D. Mimran, L. Rokach, B. Shapira, and Y. Elovici, “Mobile malware detection through analysis of deviations in application network behavior,” *Comput. Secur.*, vol. 43, pp. 1–18, 2014.
[31] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, J. B. Alis, and J. Blasco, “Dendroid: A text mining approach to analyzing and classifying code structures in Android malware families,” *Expert Syst. Appl.*, vol. 41, no. 4 PART 1, pp. 1104–1117, 2013.
[32] S. Ma, Z. Tang, Q. Xiao, J. Liu, T. T. Duong, X. Lin, and H. Zhu, “Detecting GPS Information Leakage in Android Applications,” *Glob. Commun. Conf. (GLOBECOM), 2013 IEEE*, pp. 826–831, 2013.
[33] S. Vanjire, U. Kanchan, G. Shitole, and P. Patil, “Location Based Services on Smart Phone through the Android Application,” *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 3, no. 1, pp. 4982–4987, 2014.
[34] M. Singhal and A. Shukla, “Implementation of Location based Services in Android using GPS and Web Services,” *Int. J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 237–242, 2012.
[35] McAfee Labs, “McAfee Labs Threats Report,” no. November, 2015.