Android Malware Detection: A Hybrid Approach using Machine Learning

Literature Review

Submitted By
Asadullah Hill Galib
MSSE 0718
IIT, University of Dhaka

Supervised By
Dr. B. M. Mainul Hossain
Associate Professor
IIT, University of Dhaka

Institute of Information Technology
University of Dhaka

3rd October, 2019

Author
Asadullah Hill Galib
MSSE-0718
IIT, University of Dhaka

Supervisor
Dr. B. M. Mainul Hossain
Associate Professor
IIT, University of Dhaka

Contents

1 Introduction

Android is the most ubiquitous mobile operating system nowadays [1]. It's prevalence provokes humongous growth of android malware. Researchers seek to sort out an effective strategy to defend against those malware authors. Primarily they have focused on static and dynamic analysis using machine learning to detect android malware. But, multifarious evasion techniques by the shrewd malware authors have made those approaches ineffective [2]. Yet researchers consistently aim at discovering an effectual strategy to fight against. Hybrid analysis: a fusion of static and dynamic analysis would be a good candidate for that as it prevails over the individual drawbacks of static and dynamic analysis with the cost of complexity. Recently researchers have put emphasis on this regard and revealed a lot of challenges and opportunities. This work aims at presenting a thorough review on hybrid analysis using machine learning techniques for android malware detection. It encompasses the leading researches on hybrid analysis: their contributions, contribution and limitation.

2 Literature Review

Permissions and API Calls as static features and System Calls as dynamic features are most frequently used in the existing researches. The most common dataset according to the existing researches are Drebin and Android Malware Genome Project. Besides, most researches use Google Play Store and local app stores to collect benign applications. ContagioDump, VirusTotal, VirusShare etc. sources are also used for malware samples.
Support Vector Machine (SVM) is the most frequently used machine learning algorithm in the existing research. Besides, Naive Bayes, Random Forest, J48, Logistic Regression etc. are also common in the existing researches. Accuracy, True Positive Rate (TPR), False Positive Rate are the most common evaluation metrics according to the existing researches.
In the following sub-sections, the most noteworthy literature in this regard is described:

2.1 Mobile-Sandbox (2013)

Mobile-SandBox [3] used Permissions, Services, Receivers, Intents, Potentially dangerous functions and methods as static features and investigated Native Code (Native API Calls) and Network Traffic as dynamic features to classify malware. The major contribution of it is that, it combines static and dynamic analysis first time i.e., results of static analysis are used to guide dynamic analysis. Besides it extends coverage of executed code adn uses specific techniques to log calls to native APIs. Despite the contribution, it lacks in performance as it did not provide any solid performance metrics.

2.2 Marvin (2015)

One of the state-of-the-art work in hybrid analysis, Marvin [4] employed a lot of static and dynamic features to detect android malware. It extracted Permissions, Intents, Suspicious Files, API Calls, Developer's Certificate etc. as static features and File Operations, Network Operations, Phone Events, Dynamically Loaded Code etc. as dynamic features. It used SVM and Linear Classifier (Regularized Logistic Regression) to build detection model where Linear Classifier can detect more accurately but SVM is faster comparatively. For labeled test data, Marvin's performance is sound enough as its accuracy to detect malware is 98.24 % with less than 0.04% false positive rate. But for previously unseen malware, it's accuracy is close to 90%. Besides, to avoid obsolescence of its classification model in future, it presented a retraining strategy. It also provides applicable mobile app option and outperforms other related work in performances. Though Marvin considers a lot of features, it overlooked system-level events such as System Calls : an integral part of the behavioral aspects (dynamic features).

2.3 Samadroid

Samadroid [5] presented an on-device malware detection architecture which ensures the resource efficiency by reducing memory overhead of local devices. Besides, it provides explanation to users about the behavior of application. It used a subset of Drebin's [6] features (6 out of 8) as static features and 10 predefined System Calls as dynamic features. It's accuracy is almost 98% with a false positive rate of 0.1%. Though it incorporated System Call into its feature space and outperform Drebin [6], but it used old dataset. Thereby it might fail to fight against recent malware as malware behavior changes frequently over time. It also overlooked any additional dynamic features.

2.4 HADM

Hadm [7] incorporated Deep Neural Network for feature extraction from a set of static and dynamic features. It exhibited that combining advanced features derived by deep learning with the original static and dynamic features provides consequential returns. It applied hierarchical MKL to combine different kernel learning thus further improve classification accuracy. It achieved 94.7% accuracy with a false positive rate of 1.8% where with the original features the best accuracy is 93.5%. An improvement of 1.2% with the cost of complexity.

2.5 Related Works

Kapratwar et al. [8] used Permissions and System Calls for hybrid analysis. Its performance (AUC) is significantly better for static features in comparison with dynamic features. But it used a small (200 apps) and old dataset and overlooked other static and dynamic features.

Dhanya et al.[9] used Permissions as static and API Calls as dynamic feature. Separability assessment Criteria is used for feature selection in this research. Using the 77 selected features and four different machine learning algorithms (Naive Bayes, SVM, J48 & Random Forest), they evaluated their work. Their performance regarding F-measure, precision and recall is dubitable as they used Drebin, an outdated and limited dataset. Besides they did not consider any other features except Permissions and API Calls.
Liu et al. [10] proposed a hybrid malware detecting scheme for android where Permissions and API Calls are used as static features and System Calls used as dynamic features. Their scheme's detection accuracy is from 93.33% to 99.28% according to experimental results. Though they considered only a small feature-set and their dataset is also limited.

3 Conclusion

Android malware is the key factor for the most security breaches in android operating system. Currently malware authors are sharp-witted enough to evade the typical anti viruses or obsolete approaches of malware detection. As android malware generally tries to preserve the facade of a benign application using multifarious evasion techniques, it is worthy and necessary to take a perceptive approach to defend them. Detecting android malware effectively and feasibly in advance is the biggest challenge of this fast-growing digital world. Hybrid analysis approach has the capability and can offer a sound direction on this subject.
Despite there is not so many researches are carried out in android malware detection using hybrid analysis, those few researches in this domain exhibit better performance on average than the typical static and dynamic approaches and engender a lot of opportunities. By grabbing those opportunities and overcoming the challenges ahead, hybrid analysis using machine learning would be a vanguard for android malware detection in future.
Most of the existing research dealt with some common features such as Permissions, API Calls, Intents, App Components, System Calls, File Operations, Network Operations, Phone Events, Dynamically Loaded Code etc. But it would be possible that there exists more distinguishable features to detect android malware. In this regard, Talha et al. [11] revealed many unknown characteristics of android malware, however it did not integrate any machine learning technique to detect android malware. They revealed that over-privileged permissions is one of the characteristics of malware. Besides they uncovered that malware's average number of incoming and outgoing connections, average size of download and upload, average number of INTERNET CLOSE action are distinguishable features with respect to benign applications. Looking for more discernible features might create new opportunities in android malware detection.

References

[1] Popper, B. Google announces over 2 billion monthly active devices on Android. May 17,2017; Available from: https://www.theverge.com/2017/5/17/15654454/android-reaches-2-billionmonthly- active-users.

[2] Ahn, A. How we fought bad apps and malicious developers in 2017. 1/30/2018 3/20/2018]; Available from: https://android-developers.googleblog.com/2018/01/how-we-fought-bad-appsand- malicious.html.

[3] Spreitzenbarth, M., Freiling, F., Echtler, F., Schreck, T., & Hoffmann, J. (2013, March). Mobile-sandbox: having a deeper look into android applications. In Proceedings of the 28th Annual ACM Symposium on Applied Computing (pp. 1808-1815). ACM.

[4] Lindorfer, M., Neugschwandtner, M., & Platzer, C. (2015, July). Marvin: Efficient and comprehensive mobile app classification through static and dynamic analysis. In 2015 IEEE 39th annual computer software and applications conference (Vol. 2, pp. 422-433). IEEE.

[5] Arshad, S., Shah, M. A., Wahid, A., Mehmood, A., Song, H., & Yu, H. (2018). Samadroid: a novel 3-level hybrid malware detection model for android operating system. IEEE Access, 6, 4321-4339.

[6] Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K., & Siemens, C. E. R. T. (2014, February). Drebin: Effective and explainable detection of android malware in your pocket. In Ndss (Vol. 14, pp. 23-26).

[7] Xu, L., Zhang, D., Jayasena, N., & Cavazos, J. (2016, September). Hadm: Hybrid analysis for detection of malware. In Proceedings of SAI Intelligent Systems Conference (pp. 702-724). Springer, Cham.

[8] Kapratwar, A., Di Troia, F., Stamp, M. (2017). Static and dynamic analysis of android malware. In ICISSP (pp. 653-662).

[9] T. Gireesh Kumar, K.A. Dhanya. International Journal of Recent Technology and Engineering(TM). In BEIESP (pp. 76-80)

[10] Liu, Y., Zhang, Y., Li, H., Chen, X. (2016, January). A hybrid malware detecting scheme for mobile Android applications. In 2016 IEEE International Conference on Consumer Electronics (ICCE) (pp. 155-156). IEEE.

[11] Kabakus, A. T., & Dogru, I. A. (2018). An in-depth analysis of Android malware using hybrid techniques. Digital Investigation, 24, 25-33.

[12] Enck, W., Gilbert, P., Han, S., Tendulkar, V., Chun, B. G., Cox, L. P., ... & Sheth, A. N. (2014). TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. ACM Transactions on Computer Systems (TOCS), 32(2), 5.

[13] Yan, L. K., & Yin, H. (2012). DroidScope: Seamlessly Reconstructing the OS and Dalvik Semantic Views for Dynamic Android Malware Analysis. In Presented as part of the 21st USENIX Security Symposium (USENIX Security 12) (pp. 569-584).

[14] Amos, B., Turner, H., & White, J. (2013, July). Applying machine learning classifiers to dynamic android malware detection at scale. In 2013 9th international wireless communications and mobile computing conference (IWCMC) (pp. 1666-1671). IEEE.

[15] Wu, W. C., & Hung, S. H. (2014, October). DroidDolphin: a dynamic Android malware detection framework using big data and machine learning. In Proceedings of the 2014 Conference on Research in Adaptive and Convergent Systems (pp. 247-252). ACM.