Open Android - An Android Feature Dataset for Data Scientists (Appendix)

Robert Schmicker, Frank Breitinger, and Ibrahim Baggili
University of New Haven's Cyber Forensics Research and Education Group

APPENDIX

This document is the appendix for an article and outlines the chosen literature.

TABLE I
OVERVIEW OF ARTICLES INCLUDING THEIR FEATURES UTILIZED FOR OUR WORK.

Reference	Features	Citation
[1]	Permissions, Control Flow Graphs	"In this article, we present a machine learning based system for the detection of malware on Android devices."
[2]	Permissions, API Calls, Strings, Meta Data, Opcodes, Intents	"This study summarizes the evolution of malware detection techniques based on machine learning algorithms focused on the Android OS."
[3]	Signatures, Permissions, Application Components, API Calls	"[...]we propose a novel hybrid detection system based on a new open-source framework CuckooDroid[...]"
[4]	API Calls, Permissions, System Commands	"This paper proposes and investigates a parallel machine learning based classification approach for early detection of Android malware."
[5]	Permissions, Smali Code, Intents, Strings, Components	"In this paper, we studied 100 research works published between 2010 and 2014 with the perspective of feature selection in mobile malware detection."
[6]	Permissions, Intents, Services and Receivers, SDK version API Calls, Strings	"In this paper, we present Mobile-Sandbox, a system designed to automatically analyze Android applications in novel ways[...]"
[7]	Permissions, API Calls, URI Calls	"This paper presents an approach which extracts various features from Android Application Package file (APK) using static analysis and subsequently classifies using machine learning techniques."
[8]	Components, Permissions, Intents API Calls, Strings	"In this paper, we propose DREBIN, a lightweight method for detection of Android malware that enables identifying malicious applications directlyon the smartphone."
[9]	Intents, Permissions, System Commands, API Calls	"In this chapter, we propose a machine learning based malware detection and classification methodology,with the use of static analysis as feature extraction method."
[10]	File Properties, API Calls, System Calls, JavaScript, Strings	"To discover such new malware, the SherlockDroid framework filters masses of applications and only keeps the most likely to be malicious for future inspection by anti-virus teams."
[11]	API Calls, Permissions	"In this paper, we aim to mitigate Android malware installation through providing robust and lightweight classifiers."
[12]	Permissions, API Calls	"In this paper, we present a feasibility analysis for enhancing the detection accuracy on Android malware for approaches relying on machine learning classifiers and Android applicationsâĂŹ static features."
[13]	Permissions	"In the present study, we analyze two major aspects of permission-based malware detection in Android applications: Feature selection methods and classification algorithms."
[14]	Permissions, URI Calls, Intents	"In this paper, we perform an analysis of the permission system of the Android smartphone OS[...]"
N/A*	API Calls, Smali Code	Used the decompiled smali code to "[...] link APIs to their components."

*This was our own work which is not published yet.

References

[1] J. Sahs and L. Khan, “A machine learning approach to android malware detection,” in *Intelligence and Security Informatics Conference (EISIC), 2012 European*, pp. 141–147, IEEE, 2012.

[2] B. Baskaran and A. Ralescu, “A study of android malware detection techniques and machine learning,” University of Cincinnati, 2016.

[3] X. Wang, Y. Yang, and Y. Zeng, “Accurate mobile malware detection and classification in the cloud,” *SpringerPlus*, vol. 4, no. 1, p. 1, 2015.

[4] S. Y. Yerima, S. Sezer, and I. Muttik, “Android malware detection using parallel machine learning classifiers,” in *2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies*, pp. 37–42, IEEE, 2014.

[5] A. Feizollah, N. B. Anuar, R. Salleh, and A. W. A. Wahab, “A review on feature selection in mobile malware detection,” *Digital Investigation*, vol. 13, pp. 22–37, 2015.

[6] M. Spreitzenbarth, T. Schreck, F. Echtler, D. Arp, and J. Hoffmann, “Mobile-sandbox: combining static and dynamic analysis with machine-learning techniques,” *International Journal of Information Security*, vol. 14, no. 2, pp. 141–153, 2015.

[7] V. Babu Rajesh, P. Reddy, P. Himanshu, and M. U. Patil, “Droidswan: Detecting malicious android applications based on static feature analysis,” *Computer Science & Information Technology*, p. 163, 2015.

[8] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, K. Rieck, and C. Siemens, “Drebin: Effective and explainable detection of android malware in your pocket,” in *Proceedings of the Annual Symposium on Network and Distributed System Security (NDSS)*, 2014. https://www.sec.cs.tu-bs.de/~danarp/drebin/ (last accessed 10-Feb-2017)..

[9] H. Fereidooni, V. Moonsamy, M. Conti, and L. Batina, “Efficient classification of android malware in the wild using robust static features,” 2016.

[10] L. Apvrille and A. Apvrille, “Identifying unknown android malware with feature extractions and classification techniques,” in *Trustcom/BigDataSE/ISPA, 2015 IEEE*, vol. 1, pp. 182–189, IEEE, 2015.

[11] Y. Aafer, W. Du, and H. Yin, “Droidapiminer: Mining api-level features for robust malware detection in android,” in *International Conference on Security and Privacy in Communication Systems*, pp. 86–103, Springer, 2013.

[12] D. Geneiatakis, R. Satta, I. N. Fovino, and R. Neisse, “On the efficacy of static features to detect malicious applications in android,” in *International Conference on Trust and Privacy in Digital Business*, pp. 87–98, Springer, 2015.

[13] U. Pehlivan, N. Baltaci, C. Acartürk, and N. Baykal, “The analysis of feature selection methods and classification algorithms in permission based android malware detection,” in *Computational Intelligence in Cyber Security (CICS), 2014 IEEE Symposium on*, pp. 1–8, IEEE, 2014.

[14] K. W. Y. Au, Y. F. Zhou, Z. Huang, and D. Lie, “Pscout: analyzing the android permission specification,” in *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 217–228, ACM, 2012.