Kann Malware in Android-Apps automatisch gefunden werden? (Verschiedene Analysemethoden für Android-Apps)

Cöllen, Markus
Hochschule Mannheim
Fakultät für Informatik
Paul-Wittsack-Str. 10, 68163 Mannheim

***Zusammenfassung*—An dieser Stelle steht eine kurze Zusammenfassung des Inhaltes des Dokuments.**

Inhaltsverzeichnis

1. Einleitung

2. Grundlagen

2.1. Android und Sicherheitslücken

2.2. App Store

2.3. Malware

2.3.1. Android als Ziel.

2.3.2. Klassifizierung.

3. Analysemethoden

3.1. Statische Analyse

3.1.1. Data Flow.

3.1.2. Control Flow.

3.2. Dynamische Analyse

4. FlowDroid

5. Crowdroid

6. Fazit

Eine Abkürzung = Application-to-Application (A2A) [6, S.1] [3] [8] [2] [5] [7] [4] [1] [9]

Abkürzungen

A2A Application-to-Application

Literatur

[1] Steven Arzt u.a. „FlowDroid: Precise Context, Flow, Field, Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps“. In: *SIGPLAN Not.* 49.6 (Juni 2014), S. 259–269. ISSN: 0362-1340. DOI: 10.1145/2666356.2594299. URL: http://doi.acm.org/10.1145/2666356.2594299.

[2] Steffen Bartsch u.a. „Zertifizierte Datensicherheit für Android-Anwendungen auf Basis statischer Programmanalysen.“ In: *Sicherheit*. 2014, S. 283–291.

[3] Iker Burguera, Urko Zurutuza und Simin 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*. ACM. 2011, S. 15–26.

[4] Silvio Cesare und Yang Xiang. „Classification of Malware Using Structured Control Flow“. In: *Proceedings of the Eighth Australasian Symposium on Parallel and Distributed Computing - Volume 107*. AusPDC '10. Brisbane, Australia: Australian Computer Society, Inc., 2010, S. 61–70. ISBN: 978-1-920682-88-0. URL: http://dl.acm.org/citation.cfm?id=1862294.1862301.

[5] Jianwei Ding u.a. „MGeT: Malware Gene-Based Malware Dynamic Analyses“. In: *Proceedings of the 2017 International Conference on Cryptography, Security and Privacy*. ICCSP '17. Wuhan, China: ACM, 2017, S. 96–101. ISBN: 978-1-4503-4867-6. DOI: 10.1145/3058060.3058065. URL: http://doi.acm.org/10.1145/3058060.3058065.

[6] Yu Feng u.a. „Apposcopy: Semantics-based Detection of Android Malware Through Static Analysis“. In: *Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*. FSE 2014. Hong Kong, China: ACM, 2014, S. 576–587. ISBN: 978-1-4503-3056-5. DOI: 10.1145/2635868.2635869. URL: http://doi.acm.org/10.1145/2635868.2635869.

[7] Michael Spreitzenbarth u.a. „Mobile-Sandbox: combining static and dynamic analysis with machine-learning techniques“. In: *International Journal of Information Security* 14.2 (2015), S. 141–153. ISSN: 1615-5270. DOI: 10.1007/s10207-014-0250-0. URL: https://doi.org/10.1007/s10207-014-0250-0.

[8] Lok-Kwong Yan und Heng Yin. „DroidScope: Seamlessly Reconstructing the OS and Dalvik Semantic Views for Dynamic Android Malware Analysis.“ In: *USENIX security symposium*. 2012, S. 569–584.

[9] Z. Yang und M. Yang. „LeakMiner: Detect Information Leakage on Android with Static Taint Analysis“. In: *2012 Third World Congress on Software Engineering*. 2012, S. 101–104. DOI: 10.1109/WCSE.2012.26.