Systematic literature survey

Method to conduct the systematic literature survey

The method that was used to conduct literature survey and collect different relevant papers is snowballing. The snowballing method can be done in two ways: forward and backward snowballing. In this paper, the forward snowballing was used.
To conduct a systematic literature survey, new relevant papers have to be found from the start set of papers and then be either included or excluded from the study. The method can be iterated many times until there are no new papers to be found or until the method has found enough papers to conduct the literature survey.

Snowballing

Start set of papers

To perform the snowballing method, I first received a set of five papers as a starting set. To find more papers to use as a starting set, I worked through the abstract of the received papers and wrote down a group of keyphrases that I used to find more papers as a starting point.

The keyphrases included:

1. Detecting 3rd party libraries in Android development;
2. 3rd party libraries in Android development;
3. 3rd party library detection;
4. Replacing 3rd party library in Android development;
5. API detection in Android development;
6. Evaluation of 3rd party libraries in Android;
7. Updating 3rd party libraries in Android;
8. API usage in Android.

In the end, the start set consisted of two parts: the initial set of five papers, that was sent to me and seven papers that I found from Google Scholar search, using the keyphrases I wrote down. The start set of papers are denoted P1, P2, P3 and so forth.

From the final start set I conducted 2 iterations of forward snowballing and found 22 relevant research papers.

First iteration

From the start set of 12 papers, the backward snowballing was conducted, studying the references of the 12 papers. The papers were evaluated separately (denoted P1-P12).

The papers are:

P1. Ma Z, Wang H, Guo Y, Chen X. Libradar: Fast and accurate detection of third-party libraries in Android apps. *IEEE/ACM 38th IEEE International Conference on Software Engineering Companion*, 2016

P2. Liu B, Jin H, Govindan R. Efficient privilege de-escalation for ad libraries in mobile apps. 2015. https://dl.acm.org/doi/10.1145/2742647.2742668

P3. Backes M, Bugiel S, Derr E. Reliable third-party library detection in Android and its security applications. 2016. https://dl.acm.org/doi/10.1145/2976749.2978333

P4. Huang J, Borges N, Bugiel S, Backes M. Up-to-crash: Evaluating third-party library updatability on Android. *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2019

P5. Salza P, Palomba F, Di Nucci D, De Lucia A, Ferrucci F. Third-party libraries in mobile apps. 2019. https://link.springer.com/article/10.1007/s10664-019-09754-1

P6. Concannon B. Android third party library detection. 2018. https://lib.dr.iastate.edu/creativecomponents/153

P7. Zhang Y, Dai J, Zhang X, Huang S, Yang Z, Yang M, Chen H. Detecting third-party libraries in Android applications with high precision and recall. 2018. https://ieeexplore.ieee.org/abstract/document/8330204

P8. Han H, Li R, Tang J. Identify and inspect libraries in Android applications. 2018. https://link.springer.com/article/10.1007/s11277-018-5456-4

P9. Zhang J, Beresford A R, Kollmann S A. LibID: Reliable identification of obfuscated third-party Android libraries. 2019. https://dl.acm.org/doi/abs/10.1145/3293882.3330563

P10. Lim K, Han J, Kim B, Cho S, Park M, Han S. Open-source Android app detection considering the effects of code obfuscation. 2018. http://isyou.info/jowua/papers/jowua-v9n3-5.pdf

P11. Tang Z, Xue M, Meng G, Ying C, Liu Y, He J, Zhu H, Liu Y. Securing Android application via edge assistant third-party library detection. 2018. https://www.sciencedirect.com/science/article/pii/S0167404818311301?via%3Dihub

P12. Ogawa H, Takimoto E, Mouri K, Saito S. User-side updating of third-party libraries for Android applications. 2018. https://ieeexplore.ieee.org/abstract/document/8590942

The first iteration I looked at the 12 starting papers, which all together had 495 references. Many of the references were the same, so I could exclude those references. First I looked at the references' titles and those that were irrelevant, I excluded. To decide whether a research paper based on the title might be relevant was based on the keyphrases that I wrote down in the beginning. When I had excluded all other papers from the reference list, then I looked at the abstracts of the remaining papers to see if these papers answer any of the questions that was set as a criteria that the papers would need to answer.

Those questions were:

1. What support tools are available to detect third-party libraries used in Android apps?
2. What support tools are available to assist developers if they want to change the third-party libraries in Android apps with 1) an alternative library 2) newer/stable version of the same third-party library?
3. What techniques/approaches (e.g cluster-based approach, machine-learning based, white-list based approach)these tools are using to detect/replace third-party libraries?
4. Which types (e.g ads, billing, network, social networking, game engine etc.) of third-party libraries and related issues/risks (energy consumption, security, performance, etc) can be detected/resolved by these support tools?

After this step, I excluded some of the papers again and to decide whether or not to include the paper in the study, I looked at the full paper. In the end, from the first iteration, I included in total of 9 research papers to the study.

Second iteration

The second iteration was conducted from the 9 research papers that were found in the first iteration. Again, the papers were evaluated separately (denoted P13-P21).

The papers are:

P13. Narayanan A, Chen L, Chan C K. AdDetect: Automated detection of Android ad libraries using semantic analysis. 2014. https://ieeexplore.ieee.org/abstract/document/6827639

P14. Rasthofer S, Arzt S, Bodden E. A machine-learning approach for classifying and categorizing Android sources and sinks. 2014. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.432.7534&rep=rep1&type=pdf.

P15. Li B, Zhang Y, Li J, Feng R, Gu D. AppCommune: Automated third-party libraries de-duplicating and updating for Android apps. 2019. https://ieeexplore.ieee.org/abstract/document/8668009

P16. Derr E, Bugiel S, Fahl S, Acar Y, Backes M. Keep me updated: An empirical study of third-party library updatability on Android. 2017. https://dl.acm.org/doi/abs/10.1145/3133956.3134059

P17. Salza P, Palomba F, Di Nucci D, D'Uva C, De Lucia A, Ferrucci F. Do developers update third-party libraries in mobile apps? 2018. https://dl.acm.org/doi/abs/10.1145/3196321.3196341

P18. Kula R G, German D M., Ouni A, Ishio T, Inoue K. Do developers update their library dependencies? 2018. https://link.springer.com/article/10.1007/s10664-017-9521-5

P19. Soh C, Tan H B K, Arnatovich Y L, Narayanan A, Wang L. LibSift: Automated detection of third-party libraries in Android applications. 2016. https://ieeexplore.ieee.org/abstract/document/7890569

P20. Titze D, Lux M, Schütte J. Ordol: Obfuscation-resilient detection of libraries in Android applications. 2017. https://ieeexplore.ieee.org/abstract/document/8029495

P21. Wang Y, Wu H, Zhang H, Rountev A. Orlis: Obfuscation-resilient library detection for Android. 2018. https://ieeexplore.ieee.org/abstract/document/8543426

In the second iteration, 9 papers had all together 337 references. The papers shared a lot of the same references, which I excluded from the study. Then I evaluated the references left and based on the keyphrases, I excluded the irrelevant references. After this step I looked at the remaining papers abstracts and to evaluate if these papers answer the given questions. As a last step I looked at the

whole paper and after the second iteration, I included to the study a total of 5 papers (denoted P22-P27).

The papers are:

P22. Li M, Wang W, Wang P, Wang S, Wu D, Liu J, Xue R, Huo W. LibD: Scalable and precise third-party library detection in Android markets. 2017. https://ieeexplore.ieee.org/abstract/document/7985674/

P23. Chi Z M. LibDetector: Version identification of libraries in Android applications. 2016. https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=10357&context=theses

P24. Backes M, Bugiel S, Derr E. Reliable third-party library detection in Android and its security applications. 2016. https://dl.acm.org/doi/abs/10.1145/2976749.2978333

P25. McDonnel T, Ray B, Kim M. An empirical study of API stability and adoption in the Android ecosystem. 2013. https://ieeexplore.ieee.org/abstract/document/6676878/

P26. Mojica I J., Adams B, Nagappan M, Dienst S, Berger T, Hassan A E. A large-scale empirical study on software reuse in mobile apps. 2014. https://ieeexplore.ieee.org/abstract/document/6663589

Analysis

After two iterations of forward snowballing, I have 27 papers including the 12 initial papers. The following is the analysis of the 27 papers to answer the questions mentioned before.

Denotation	Presents a tool?	Type of tool	Approach	Type of library	Link
P1	Yes	Detection	Cluster-based	Any	https://github.com/pkumza/LibRadar
P2	Yes	Detection	Machine-learning based	Ad	Pedal
P3	No	-	-	-	-
P4	No	-	-	-	-
P5	No	-	-	-	-
P6	No	-	-	-	-
P7	Yes	Detection	Class-dependency based	Any	https://github.com/yuanxzhang/LibPecker
P8	No	-	-	-	-
P9	Yes	Detection	Locality-Sensitive Hashing	Any	https://github.com/ucam-cl-dtg/LibID
P10	No	-	-	-	-
P11	Yes	Detection	Machine-learning based	Any	PanGuard
P12	No	-	-	-	-
P13	Yes	Detection	Machine-learning based	Ad	-
P14	Yes	Detection	Machine-learning based	Any	https://github.com/secure-software-engineering/SuSi
P15	Yes	Updating	White-list based	Any	-
P16	No	-	-	-	-
P17	No	-	-	-	-
P18	No	-	-	-	-
P19	Yes	Detection	Package-dependency based	Any	-
P20	Yes	Detection	Iterative heuristic approach	Any	https://github.com/milux/ordol

Denotation	Presents a tool?	Type of tool	Approach	Type of library	Link
P21	Yes	Detection	Using interprocedural code features and similarity digests	Any	https://github.com/presto-osu/orlis-orcis
P22	Yes	Detection	Classification technique	Any	https://github.com/IIE-LibD/libd
P23	Yes	Version Identification	White-list based	Any	https://github.com/zchi88/LibDetector
P24	No	-	Based on class hierarchy information	Any	-
P25	No	-	-	-	-
P26	No	-	-	-	-

References

Kitchenham B. Guidelines for performing systematic literature reviews in software engineering. *EBSE Technical Report*, 2007.

Wohlin C. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. 2014. https://dl.acm.org/doi/10.1145/2601248.2601268

Ma Z, Wang H, Guo Y, Chen X. Libradar: Fast and accurate detection of third-party libraries in Android apps. *IEEE/ACM 38th IEEE International Conference on Software Engineering Companion*, 2016

Liu B, Jin H, Govindan R. Efficient privilege de-escalation for ad libraries in mobile apps. 2015. https://dl.acm.org/doi/10.1145/2742647.2742668

Backes M, Bugiel S, Derr E. Reliable third-party library detection in Android and its security applications. 2016. https://dl.acm.org/doi/10.1145/2976749.2978333

Huang J, Borges N, Bugiel S, Backes M. Up-to-crash: Evaluating third-party library updatability on Android. *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2019

Salza P, Palomba F, Di Nucci D, De Lucia A, Ferrucci F. Third-party libraries in mobile apps. 2019. https://link.springer.com/article/10.1007/s10664-019-09754-1

Concannon B. Android third party library detection. 2018. https://lib.dr.iastate.edu/creativecomponents/153

Zhang Y, Dai J, Zhang X, Huang S, Yang Z, Yang M, Chen H. Detecting third-party libraries in Android applications with high precision and recall. 2018. https://ieeexplore.ieee.org/abstract/document/8330204

Han H, Li R, Tang J. Identify and inspect libraries in Android applications. 2018. https://link.springer.com/article/10.1007/s11277-018-5456-4

Zhang J, Beresford A R, Kollmann S A. LibID: Reliable identification of obfuscated third-party Android libraries. 2019. https://dl.acm.org/doi/abs/10.1145/3293882.3330563

Lim K, Han J, Kim B, Cho S, Park M, Han S. Open-source Android app detection considering the effects of code obfuscation. 2018. http://isyou.info/jowua/papers/jowua-v9n3-5.pdf

Tang Z, Xue M, Meng G, Ying C, Liu Y, He J, Zhu H, Liu Y. Securing Android application via edge assistant third-party library detection. 2018. https://www.sciencedirect.com/science/article/pii/S0167404818311301?via%3Dihub

Ogawa H, Takimoto E, Mouri K, Saito S. User-side updating of third-party libraries for Android applications. 2018. https://ieeexplore.ieee.org/abstract/document/8590942

Narayanan A, Chen L, Chan C K. AdDetect: Automated detection of Android ad libraries using semantic analysis. 2014. https://ieeexplore.ieee.org/abstract/document/6827639

Rasthofer S, Arzt S, Bodden E. A machine-learning approach for classifying and categorizing Android sources and sinks. 2014. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.432.7534&rep=rep1&type=pdf.

Li B, Zhang Y, Li J, Feng R, Gu D. AppCommune: Automated third-party libraries de-duplicating and updating for Android apps. 2019. https://ieeexplore.ieee.org/abstract/document/8668009

Derr E, Bugiel S, Fahl S, Acar Y, Backes M. Keep me updated: An empirical study of third-party library updatability on Android. 2017. https://dl.acm.org/doi/abs/10.1145/3133956.3134059

Salza P, Palomba F, Di Nucci D, D'Uva C, De Lucia A, Ferrucci F. Do developers update third-party libraries in mobile apps? 2018. https://dl.acm.org/doi/abs/10.1145/3196321.3196341

Kula R G, German D M., Ouni A, Ishio T, Inoue K. Do developers update their library dependencies? 2018. https://link.springer.com/article/10.1007/s10664-017-9521-5

Soh C, Tan H B K, Arnatovich Y L, Narayanan A, Wang L. LibSift: Automated detection of third-party libraries in Android applications. 2016. https://ieeexplore.ieee.org/abstract/document/7890569

Titze D, Lux M, Schütte J. Ordol: Obfuscation-resilient detection of libraries in Android applications. 2017. https://ieeexplore.ieee.org/abstract/document/8029495

Wang Y, Wu H, Zhang H, Rountev A. Orlis: Obfuscation-resilient library detection for Android. 2018. https://ieeexplore.ieee.org/abstract/document/8543426

Li M, Wang W, Wang P, Wang S, Wu D, Liu J, Xue R, Huo W. LibD: Scalable and precise third-party library detection in Android markets. 2017. https://ieeexplore.ieee.org/abstract/document/7985674/

Chi Z M. LibDetector: Version identification of libraries in Android applications. 2016. https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=10357&context=theses

Backes M, Bugiel S, Derr E. Reliable third-party library detection in Android and its security applications. 2016. https://dl.acm.org/doi/abs/10.1145/2976749.2978333

McDonnel T, Ray B, Kim M. An empirical study of API stability and adoption in the Android ecosystem. 2013. https://ieeexplore.ieee.org/abstract/document/6676878/

Mojica I J., Adams B, Nagappan M, Dienst S, Berger T, Hassan A E. A large-scale empirical study on software reuse in mobile apps. 2014. https://ieeexplore.ieee.org/abstract/document/6663589