PermDroid : Prévenir les failles de sécurité liées aux permissions dans les applications Android

Mohammed El Amin TEBIB ∗ , Pascal André † , Mariem Graa ‡ , Oum-El-Kheir Aktouf ∗
∗Univ. Grenoble Alpes, Grenoble INP, LCIS - Email: mohammed-el-amin.tebib, oum-el-kheir.aktouf@univ-grenoble-alpes.fr
† Univ. of Nantes, LS2N - Email: pascal.andre@ls2n.fr
‡ CNAM - Nantes - Email: mariem.graa@gmail.com

***Contexte* et motivations-** Malgré divers travaux de recherches menés pendant la dernière décennie sur la sécurité des applications Android, les problèmes de sécurité persistent. En 2020 et début 2021, les statistiques restent très significatives au sujet du nombre de failles de sécurité liées à ce type d'applications (1148 failles, dont 791 liées aux privilèges) selon le CVE[1]. Une autre étude expérimentale récente [2] effectuée sur des applications Android open source (574 référentiels github) a montré que les problèmes de sécurité liés aux autorisations sont toujours un phénomène fréquent dans les applications Android. Ce qui conduit à des attaques de sécurité très connues comme l'escalade de privilèges et l'exploitation des données privées.

Les autorisations sur Android sont manipulées par les développeurs à travers le concept de permissions et sont accordées par l'utilisateur pendant l'exécution de l'application. Une permission sera définie dans le code de l'application pour lui donner l'autorisation nécessaire d'utiliser une ressource exposée par le système. Pour manipuler ces permissions (connaître leur signification, comment les utiliser et pour quels objectifs), Google met à disposition des développeurs une documentation officielle expliquant leur utilisation[2]. Cependant, en raison des changements continus du nombre et des spécifications des autorisations, cette documentation devient rapidement obsolète et difficilement lisible pour les développeurs (qui sont souvent des développeurs tiers- c'est-à-dire non spécialistes de la sécurité).

***Problématique*-** Après une étude approfondie (et qui se poursuit toujours) des solutions proposées dans la littérature ([3], [4]) pour faire face à cette situation qui entraîne différents problèmes et risques de sécurité. Ce travail traite principalement les limites suivantes : *1) Approche*. La plupart des approches sont statiques. Ce qui donne une bonne couverture d'analyse mais avec moins de précision. Il manque aussi des outils d'aide pour détecter des erreurs pendant le développement et pour assurer la vérification des propriétés de sécurité à travers une approche rigoureuse formelle. Nous traitons ici en particulier les propriétés liées aux permissions des applications. *2) Propriétés de sécurité.* Concernant les permissions, on cible deux catégories de propriétés : i) Des propriétés génériques liées aux conflits de nommage de permissions, aux fuites liées à l'invocation incorrecte des composants ou l'escalade des privilèges. ii) Les propriétés liées aux problématiques de sur-privilèges : la plupart des approches proposées sont statiques et analysent seulement les permissions déclarées au niveau du fichier de configuration manifest.xml et les classes Java. Ces approches n'assurent pas une analyse des permissions utilisées dans plusieurs autres niveaux tels que les APIs natives (implémentées en C/C++), les permissions accordées à l'exécution, les permissions accordées à travers l'utilisation de Java Reflection.

Contributions- Ce projet contribue à l'implémentation d'un outil d'assistance ***(PermDroid [1])***[3] pour vérifier les propriétés de sécurité durant le développement des applications Android. En se basant sur les limites de sécurité décrites plus haut, le but est que l'approche d'analyse implémentée par l'outil arrive à :

1) Couvrir des permissions utilisées dans tout type d'API utilisable dans une application Android (API java, API native, etc.). 2) Une approche d'analyse dynamique pour analyse de la réflexion Java. L'outil est développé sous forme d'un plugin et intégré dans les environnements de développement comme Android Studio[4].

REFERENCES

[1] Tebib, M. A., André, P., Aktouf, O., & Graa, M. Assisting Developers in Preventing Permissions Related Security Issues in Android Applications. *EDCC Workshops* (2021).

[2] Scoccia, G. L., Peruma, A., Pujols, V., Malavolta, I., & Krutz, D. E. Permission issues in open-source Android apps: An exploratory study. *In 2019 19th International Working Conference on Source Code Analysis and Manipulation (SCAM)*. pp. 238-249, September 2019.

[3] Vidas, T., Christin, N., & Cranor, L. Curbing Android permission creep. *In Proceedings of the Web (Vol. 2, pp. 91-96)*. May 2011.

[4] Kaplan, Avi & Maehr, Martin L. The Contributions and Prospects of Goal Orientation Theory. *Educational Psychology Review* 19(2), pp. 141-184. June 2007.

[1] https://www.cvedetails.com/product/19997/Google-Android.html?vendor_i

[2] https://developer.android.com/studio

[3] https://github.com/tebmed

[4] https://developer.android.com/studio