COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

CentraleSupélec

Titre : Les difficultés de la rétro-ingénierie Android: de l'analyse large échelle au dé-brouillage dynamique

Mots clés : Android, Analyse de Maliciels, Ingénierie Inverse, Chargement de Classe, Brouillage de Code

Résumé : La place croissante des téléphones mobiles dans notre vie quotidienne en font une cible de choix pour les acteurs malveillants. Cette menace rend l'analyse d'application cruciale pour déterminer s'il s'agit ou non d'un maliciel et de son impact sur l'utilisateur s'il s'agit bien d'une application malveillante.

Cette thèse explore les difficultés liées à l'ingénierie inverse d'applications Android. Dans un premier temps, elle reprend un effort de la communauté qui a identifié les contributions entre 2011 et 2017 portant sur l'analyse statique d'applications mobiles et propose une méthode pour évaluer la réutilisabilité des outils associés. Une étude poussée des échecs lors de l'exécution de ces outils montre que 54.55% d'entre eux ne sont plus utilisables. Elle modélise ensuite le processus de chargement des classes utilisées par une application et présente une méthode de brouillage basée sur les différences entre ce modèle et celui utilisé par des outils d'analyses tels que Androguard ou Flowdroid. Enfin, elle propose une approche consistant à encoder les résultats d'une analyse dynamique dans une nouvelle application valide pour permettre aux outils existants d'analyser des applications faisant usage de chargement de code dynamique.

Title : The Woes of Android Reverse Engineering: from Large Scale Analysis to Dynamic Deobfuscation

Keywords: Android, Malware Analysis, Reverse Engineering, Class Loading, Code Obfuscation

Abstract: The growing importance of smartphones in our daily lives makes them a prime target for malicious actors. This makes our ability to analyse an application critical, as it allows us to determine if an application is malware and its impact on the user.

This thesis explores the difficulties of reverse engineering an Android application. First, we pursue a community effort that identified contributions between 2011 and 2017 about static analysis of Android applications, and we propose a method to evaluate the reusability of the associated tools. An extensive analysis of the execution failures of those tools shows that 54.55% of them are no longer usable. Then, we model the mechanism that loads the classes used by an application and present an obfuscation method based on the discrepancies between this model and the one used by analysis tools like Androguard and Flowdroid. Finally, we propose an approach that consists of encoding the result of a dynamic analysis within a new valid application, allowing existing tools to analyse applications that rely on dynamic code loading.