TECHNIQUES FOR DETECTION OF MALWARE IN ANDROID APPLICATIONS

Varna Priya D, Research Scholar, Dept. of ECE, PSGCT
Visalakshi P, Professor, Dept. of ECE, PSGCT

The father of ubiquitous computing Mark Weiser, described that "classical computers will be replaced by lightweight, smart, distributed and networked devices that are incorporated into everyday objects and activities", which is achieved by today's smartphones. For many people, smartphones play a predominant role in their day-to-day lives. The smartphones are used for wide range of activities from storing contacts, sending messages to sensitive applications like mobile banking. There are several operating systems that serve as a platform for these smartphones and Android is one such widely used operating system.

Android holds a majority of the smartphone operating system's market share, accounting to nearly 76% of the global share. One vital reason behind the popularity of the Android operating system is its open source nature. Android also allows a greater range of customization options to its consumers. This popularity of Android has attracted the attention of malware writers. As per a report given by Kaspersky, 35, 03,952 malicious installation packages are detected in 2019, in which nearly 70,000 packages are Trojans that are developed targeting the mobile banking applications. Most malwares affecting the Android devices are mainly from the applications that are downloaded from third-party markets.

Google makes every effort to keep its official market "Play Store", free of malwares. Despite their best efforts, eradication of malware from Google's Play Store is not yet fully accomplished. With the enormous amount of malware emerging every day, the existing signature based detection is not efficient. For the reasons stated above, the detection of malware in Android applications has gained significance. This research focuses on few methods and tactics that can be used in the detection of malware in Android applications, with the objective of achieving a higher True Positive Rates and reduced False Positive Rates.

Android operating system ensures the security of the devices that are powered by them with the use of sandboxes. If an application requires accessing the features or data of the other application, then explicit permissions are required to do so. Hence, permissions requested by the applications serve as one salient feature in detection of Android malware, when static analysis technique is used. With dynamic analysis the system calls invoked by the applications serve as a key feature.

The machine learning and deep learning models are contemporary methods that are used in many data mining problems. The Android applications can be analysed either using static or dynamic analysis. The performance of any classification algorithm can be improvised by feeding the right or more appropriate features to the algorithms. Hence, feature selection techniques also play a vital role in the detection models. The research thus focuses on feature selection techniques in addition to the usage of machine and deep learning methodologies. Standard metrics like TPR, FPR, and Accuracy are used for comparison of the proposed techniques.

The work used malware samples from two bench-marked datasets namely the Android Adware and General Malware (AAGM) dataset and the Android Malware Dataset (AMD). The benign samples are downloaded from Google Play Store and the authenticity is verified with the help of VirusTotal.

The first contribution of the research suggests usage of modified Support Vector Machines (SVM) Kernels, termed as "Optimized SVM" (o-SVM), where hybridization of kernels is suggested as an alternative. The proposed technique is compared with few existing machine learning algorithms, to understand the performance of the proposed method.

The second work focuses on feature selection. An algorithm called K-Nearest Neighbour based Relief (KNN-R), was suggested as an improvised version of the popular Relief algorithm, to select features that are more pertinent, and can be used to help improve the detection of Android malware. The proposed algorithm is found to achieve better performance with both static and dynamic features.

The third work is based on dynamic analysis, where several sub-sequences of the system calls generated are extracted to construct feature vectors. The usages of frequency of system calls invoked by the applications are not efficient, as any action requires invocations of few contiguous system calls. The depth of the sub-sequence is varied and the performance of the model is studied.

The final work focuses on static analysis. Permissions extracted from the Android's manifest file are one predominant feature that is used Android malware detection. The proposed work suggests a technique called "Permission Grading System (PGS)", which, identifies permissions that are unique to benign and malware samples and eliminates those permissions that are common in both. The proposed PGS system thus provides a way of identifying those permissions that are more useful in detection of malware.