CSE3502 – INFORMATION SECURITY MANAGEMENT

Review 1

ANDROID MALWARE ANALYSIS

Prof. In-Charge:

Dr. Vimala Devi

Team Details:

Harshit Mishra (19BCE0799)

Alokam Nikhitha (19BCE2555)

Shreeyam Sharma (19BCE2700)

Abstract:

Android is an open-source Operating System with more than a billion users. The amount of sensitive information produced be these technologies are rapidly increasing, which attracts a large number of audiences to develop tools and techniques to acquire that information or to disrupt the device's smooth operation. Despite several solutions being able to guarantee an adequate level of security, day by day the hacker's skills continues to grow, so it remains a permanent challenge for security tools developers to ensure the security of an android powered device.

As a response, several members of the research community are using artificial intelligence tools for android security, particularly machine learning techniques to classify between healthy or malicious android application.

In this project, we will implement a static framework and machine learning to do this classification.

Problem Statement and Objective:

Android is an open-source operating system for mobile devices, televisions automobiles and smart watches with more than a billion users. Therefore, it opens a wide array of attack vectors targeting the user information.

For the protection of the information and devices, android has several security mechanisms; the most relevant are: a sandbox environment at the kernel level to prevent access to the file system and other resources; an API of permissions that controls the privileges of the applications in the device; security mechanisms at the applications development level; and a digital distribution platform (Google play store), where the processes are implemented to limit the dissemination of malicious code.

Each application is compiled in an Android Application Package [APK] file, which includes the code of the application in ". dex" files, resources and the AndroidManifest.xml file. This latter is an important element, since it provides most of the information of the security features and configuration of each application. It also includes the information of the API regarding permissions, activities, services, content providers and the receiving broadcasts.

There are several tools and techniques for the analysis of threats for this operating system. Between the most representative, we have static analysis and dynamic analysis.

Static Analysis:

Static analysis is a technique that assesses behaviour in the source code, the data, or the binary files without the direct execution of the application. Its complexity has increased due to the experience that cybercriminals have gained in the development of applications. However, it has been demonstrated that it is possible to avoid this using obfuscation technique.

Dynamic Analysis:

Dynamic analysis is a set of methods that studies the behaviour of the malware in execution through gesture simulations. In this technique, the process in execution, the user interface, the network connections and sockets opening are analysed. Alternatively, there already exist some technique to avoid the processes performed by dynamic analysis, where the malware has the capacity to detect sandbox-like environments and to stop its malicious behaviour.

Literature Survey:

Paper	Problem and Objective	Proposed Methodology	Limitations
Significant Permission Identification for Machine Learning-Based Android Malware Detection **Jin Li, Lichao Sun, Qiben Yan, Zhiqiang Li, Witawas Srisa-an and Heng Ye** July-2018 IEEE Transaction on Industrial Informatics	The disturbing development pace of noxious applications has turned into a major issue that interferes with the prosperous versatile environment. A new report shows that a new vindictive application for Android is presented each 10 s. To battle this genuine malware crusade, we really want an adaptable malware identification approach that can successfully and effectively recognize malware applications. Various malware identification instruments have been created, including framework level and organization level methodologies. Notwithstanding, scaling the discovery for a huge heap of applications stays a difficult assignment. In this paper, we present Significant Permission IDentification (SigPID), a malware recognition framework in light of authorization utilization investigation to adapt	In this work, they have proposed a SigPID, a malware recognition framework in light of authorization utilization investigation to adapt to the fast expansion in the quantity of Android Malware. SigPID stands for Significant Permission Identification. SigPID utilizes machine-learning-based classification methods to classify different families of malware and benign applications. Their evaluation found that only 22 permissions among the applications stand significant. They then compared the performance of their approach, using only these 22 permissions, against a baseline approach that analyses all permissions. The results indicate that when a support vector machine is used as the classifier, they achieved an accuracy over 90%.	The approach needed high end processors, large memory space and Graphical processing unit. Since most of the common computers lack these specifications, the approach is not as implementable and is rather subject to hardware. The process is redundantly slow due the extensive use of SigPID framework which although increases the accuracy by leaps but alters the pace of detection.

	to the fast expansion in the quantity of Android malware. Rather than separating and breaking down all Android consents, we foster three degrees of pruning by mining the authorization information to recognize the main consents that can be viable in recognizing harmless and pernicious applications		
MEGDroid: A model-driven event generation framework for dynamic android malware **Hayyan Hasan and Behrouz Tork Ladani** July-2021 Information and Software Technology	The tremendous growth of Android Malware in recent years is a strong motivation for the vast endeavour in detection and analysis of malware apps. A prominent approach for this purpose is dynamic analysis in which providing complex interactions with the samples under analysis is a need. Event generation tools are almost used to provide such interactions, but they have deficiencies for effective malware analysis.	They have proposed a MEGDroid, a Model Driven Engineering (MDE) framework in which malware-related information is automatically extracted and represented as a domain-specific model. This model, then is used to generate appropriate events for malware analysis using model-to-model and model-to-code transformations. The proposed model-driven artifacts also provide required facilities to put the human in the loop for properly taking his/her knowledge into account.	There are quite a few limitations for achieving the gola due to the anti-static and anti-dynamic analysis techniques that are usually used by malware to hide their information. The obfuscation approach can be used to by pass this detection process. Thus, if the hacker is smart enough the process of MEGDroid detection is not sufficient to detect the malware.
An Android Malicious Code Detection Method Based on Improved DCA algorithm	Android Malicious Code has increased dramatically and the technology of reinforcement is increasingly	This paper proposed a Dendritic Cell Algorithm (DCA), which is an Android malware algorithm that has a higher	The working of proposed system works only if they can access the Android Packaging (APKs) of a

Chundong Wang, Zhiyuan Li, Liangyi Gong, Xiulian Mo, Hong Yang and Yi Zhao February-2017 Entropy-Based Applied Cryptography and Enhanced Security for Future IT Environments	powerful. Due to the development of code obfuscation and polymorphic deformation technology, the current android malicious code static detection method whose feature selected is the semantic of application sources code cannot completely extract malware's code features. The Android malware static detection methods whose features used are only obtained from the AndroidManifest.xml files are easily affected by useless permissions.	detection rate, does not need to modify the system, and reduces the impact of code obfuscation to a certain degree. This algorithm is applied to an Android malware detection method based on oriented Dalvik disassembly sequence and application interface (API) calling sequence. Through the designed experiments, the effectiveness of this method is verified for the detection of Android Malware. This is a dynamic implementation approach which studies the execution phase of an application and classifies them as safe or not based on it.	particular application. It requires a large number of APKs to deliver optimal working. With less than 400 APKs the accuracy of model stood at a mere 92%. With around 750 APKs it went to a count of 97%. Hence we need large number of APKs for it to work accurately.
Machine Learning Aided Android Malware Classification **N Milosevic, A Dehghantanha and k Choo** Februray-2017 White Rose university Consortium	Malware have been used as a means for conducting cyber attacks for decades. With adoption of smartphones, which stores lots of private and confidential information, made them an important target for malware developers. Android as the dominant mobile operating system has always been an interesting platform for malware developers and lots of Android malware	In the work, they have proposed two Machine Learning based approaches for static analysis of the mobile applications: one based on permissions, while the other based on source code analysis that utilizes a bag of word representation model. Their source-code based classification achieved F-score of 95.1%, while the approach that used	Their approach had few limitations. For permission-based approach they reported an F-score of 87% for single machine learning algorithm. This means there are chances that some malware loaded applications were not classified as such and some benign applications are classified as malicious. Also, the False Negative rates in

	species are infecting vulnerable users everyday which make manual malware forensics would assist cyber forensics investigators in their fight against malicious programs.	permission names only performed with F-measure of 89%. Their approach provides a method for automated static code analysis and malware detection with high accuracy and reduces smartphone malware analysis time.	their case is high, which means that the detection rate of malware applications was low and this possessed a potential vulnerability in classification due to this alarming false negative rates.
AdDroid: Rule-Based machine Learning Framework for Android Malware Analysis **Anam Mehtab, Waleed Bin Shahid and Tahreem Yaqoob** August-2019 Mobile Networks and Applications	Recent years have witnessed huge growth in Android malware development. Colossal reliance on Android applications for day to day working and their massive development dictates for an automated mechanism to distinguish malicious applications from benign ones. A significant amount of research has been devoted to analysing and mitigating this growing problem; however, attackers are using more complicated techniques to evade detection.	This paper proposed a framework named AdDroid; for analysing and detecting malicious behaviour in Android Applications based on various combinations of artefacts called rules. The artefacts represent actions of an Android application such as connecting to the Internet, uploading a file to a remote server or installing another package on the device etc. AdDroid employs an ensemble-based machine learning technique where Adaboost is combined with traditional classifiers in order to train a model founded on static analysis of Android applications that is capable of recognizing malicious applications. Feature Selection and extraction	The dataset used for the approach was very biased towards malicious applications with a ratio approximately reaching 2:3. This suggest that the results of accuracy is also biased towards malicious applications. This further suggests that the rate of False positive data is going to be high because the model is more fitted to those applications. The model was able to detect applications on the basis of the permissions they have, this limits the scope and scalability of approach and applications in real time cannot be classified using this approach.

		techniques were used to get the most distinguished Rules. The proposed model was created using a dataset comprising 1420 Android Applications with 910 malicious and 510 being benign.	