iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VI **Month of publication:** June 2020

DOI: http://doi.org/10.22214/ijraset.2020.6160

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VI June 2020- Available at www.ijraset.com

On Malware Detection on Android Smartphones

Eman Shalabi[1], Ahmed Moustafa[2], Walid Khedr[3]
[1, 2, 3]*Zagazig University*

Abstract: ***Smartphones plays an important role in our daily life. They have become a significant part of our daily life. They can be used in many fields such as online banking, online learning, social networking, web browsing, etc. The large increase in the use of smartphones leads to a large increase in generating mobile malware. In this paper, we discuss various mobile malware types and datasets used for mobile malware detection process. We also survey various mobile malware detection techniques.***
General Terms: Malware, Malware Types, Android, Machine Learning and Android Malware Detection Approaches.
Keywords: Malware, Malware Types, Android, Android Malware Detection, Malware Dataset, Static Analysis, Dynamic Analysis and Hybrid Analysis.

I. INTRODUCTION

Now-days smart-phones are becoming very famous all around the world. As the study says, among all platforms, Android is the widely used platform. With the rising pervasiveness of using these mobile platforms in delicate applications, there is a problems associated with malware targeted at mobile devices.[1]

A mobile malware can be defined as malicious software. This software targets mobile operating system. Mobile Malware could be any code which is added, changed or removed from any mobile application in order to damage or harm the function of the intended system [2].

Machine Learning plays an important role in malware detection process. It can be used in order to improve malware detection accuracy.

In [3], to identify whether an Android application is malicious or not, the authors presented an ML system based on machine learning techniques and features extracted from Android API calls and system calls. This system achieved 96.66 percent detection rate. They used more than one machine learning classifier Random Forest 10, Random Forest 50, Random Forest 100, J.48, Naïve Bayes, Simple Logistic, BayesNet TAN, BayesNet K2, SMO PolyKernel, SMO NPolyKernel, IBK 1, IBK 3, IBK 5 and IBK 10. Their dataset consists of a total of 7,520 apps, 3,780 for training and 3,740 for testing. The malicious dataset consists of totaling 4,552 samples, collected from the android MalGenome project and a torrent file acquired from VirusShare. The benign dataset consists of 2,968 applications, collected from the AndroidPIT by using their own developed crawler system and Virustotal to scan them.

In [4] The authors presented a machine learning approach for android malware detection based on network traffic. They used seven network traffic features which are Average packet size, Average Number of Packets Sent per Flow, Average Number of Packets Received per Flow, Average Number of Bytes Sent per Flow, Average Number of Bytes Received per Flow, Ratio of Incoming to outgoing Bytes, and Average Number of Bytes Received per Second. These features extracted from two datasets which are Drebin dataset and Contagiodumpset. Their dataset consists of 700 samples including 500 samples for benign applications and 200 samples for malware applications (100 samples from Drebin dataset and 100 samples from Contagiodumpset). This research evaluated J48 decision tree machine learning algorithm for the detection of android malware which achieved a 98.4% accuracy for Drebin dataset 97.6% for Contagiodumpset dataset.

In [5] the authors proposed 'BehaveYourself!' Which is a machine learning-based Android application used for classifying and discriminating a trusted Android application by a malicious one. BehaveYourself detects malware using opcode-based features. Move, Jump, Packed-Switch, Sparse-Switch, Invoke and If are the 6 opcode features used in this research. Their dataset consists of 5,560 malware samples from Drebin dataset and 5,560 benign samples from Google play store. The model is built with the J48 decision tree classifier using the 10-fold cross validation and achieved a precision of 94.9%.

The remainder of this study is organized as follows. Section 2 discusses various malware types. Section 3 illustrates some of mobile malware datasets. Section 4 discusses the detection techniques of mobile malware.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VI June 2020- Available at www.ijraset.com

II. MALWARE TYPES

A mobile malware can be defined as malicious software. This software targets mobile operating system. Mobile malware may be any code that is inserted, altered or removed from any mobile application to harm or impair the intended system's operation. There are many different malware types such as Adware, Bot, Bug, Spyware, Virus, Trojans, Worm, Rootkit and Ransomware. Most mobile malware is designed to disable or harm a mobile device which allows a malicious user to remotely control the mobile device or to steal personal data stored on the mobile device. Mobile malware has threatened smartphones for many years.

A. Adware

Adware which stands for advertising-supported software that automatically delivers advertisements. Common examples of adware are Pop-up ads and advertisements on websites. Often, software and applications offer free versions which bundle adware. Most adware designed is sponsored or authored by advertisers and serves as a revenue generating tool, but some adware is designed solely to deliver advertisements and it is not uncommon for adware to come bundled with spyware that can track user activity and steal sensitive information. Due to the added capabilities of spyware, adware/spyware bundles are significantly more serious and dangerous than adware on its own.

B. Bot

Bots are software programs. This software program was created to execute specific operations automatically. Although some bots are designed and created for relatively harmless purposes such as video games, internet auctions, online competitions, etc., use of bots for malicious activities is becoming more and more common. Bots can be used in botnets (collections of computers controlled by third parties) for the purpose of DDoS attacks, rendering of spam bots advertisements on websites, web spiders which scraping server data, and for the distribution of malware on download sites. Through CAPTCHA tests which verify users as human, websites can guard against bots.

C. Bug

In the context of software, a bug is a flaw produces an undesired outcome. Usually these flaws are the result of human error, and typically exist in a program's source code or compilers. Minor bugs only slightly affect the behavior of a program and thus can go on for long periods of time before it is discovered. It may cause more serious bugs to crash or freeze. Security bugs are the most dangerous kind of bug and can allow attackers to bypass user authentication, override access privileges, or steal data. Bugs can be prevented with developer education, quality control, and code analysis tools.

D. Spyware

Spyware is a malware type. Spyware functions and operates by spying on user activity without their knowledge. Spying capabilities can include monitoring activities, collecting key strokes, collecting data (account information, logins, financial data, etc.). Spyware also often has extra features, ranging from modifying device or application security settings to interfering with network connections. Spyware spreads by exploiting software vulnerabilities, bundling itself with legitimate software, or in Trojans.

E. Virus

A virus is a form of malware which can copy itself and spread to other computers. Viruses frequently spread to other computers by attaching to different programs and executing code when a user launches one of those infected programs. Viruses can also spread vulnerabilities in Web applications through script files, documents, and cross-site scripting. Viruses can be used to steal information, harm host computers and networks, build botnets, steal money, and make advertisements and more.

F. Worm

Computer worms rank among the most common malware types. They spread through computer networks by exploiting vulnerabilities of the operating system. Worms typically cause harm to their host networks by bandwidth consumption and overloading of web servers. Computer worms may also have payloads destroying host computers. Payloads are pieces of code written to perform actions on computers that are infected beyond simply spreading the worm. Commonly, payloads are designed to steal data, remove files, or create botnets. Computer worms can be classified as a computer virus type but there are several features that distinguish computer worms from regular viruses. A big difference is that computer worms have the ability to reproduce themselves and spread independently while viruses rely on spreading human activity (running a program, opening a file, etc.). Worms also spread via the sending to users' contacts of mass emails with infected attachments.

G. Trojan Horse

A Trojan horse, commonly known as Trojan horse, is a type of malware that disguises itself as a normal file or program that tricks downloading and installing malware on users. A Trojan can give a remote access to an infected computer from a malicious party. Once an attacker accesses an infected computer, the attacker can steal data (logins, financial data, and even electronic money), install more malware, modify files, monitor user activity (screen watching, key logging, etc.), use the computer in botnets, and anonymize the attacker's internet activity.

H. Rootkit

Rootkit is a form of malicious software designed to access or monitor a device remotely without users or security programs detecting it. If a rootkit is installed, the malicious party behind the rootkit may execute files remotely, access / steal information, change device settings, alter software (especially any security software that might detect the rootkit), install concealed malware or control the machine as part of a botnet. Because of their stealthy operation, prevention, detection, and removal of rootkits can be difficult. Typical security products are not effective in detecting and removing rootkits, because a rootkit continually hides its presence. As a Utility As a result, rootkit detection relies on manual methods such as monitoring computer behavior for irregular activity, signature scanning, and storage dump analysis. By regularly patching vulnerabilities in software, applications, and operating systems, updating virus definitions, avoiding suspicious downloads, and performing static analysis scans, organizations and users can protect themselves from rootkits.

I. Ransomware

In 2014 Ransomware infects network Android. It is a type of malware which essentially holds a captive computer system while demanding a ransom. The malware prevents and limits user access to the device by either encrypting files on the hard drive or locking down the machine and showing messages that are intended to compel the user to pay the malware maker to remove the restrictions and get back access to their computer. Ransomware typically spreads through a downloaded file, or through some other vulnerability in a network service, like a benign and normal computer worm ending up on a computer.

Despite these types of malware, they do have a great difference in how computers are spread and infected; they can all produce similar symptoms.

1) Increase the usage of CPU
2) Slow computer machine or web browser speeds
3) Problems of network connection
4) Freezing or crashing
5) Appearance of modified or deleted files
6) Appearance of strange files, programs, or desktop icons
7) Programs running, turning off, or reconfiguring themselves (malware may also reconfigure or switch off antivirus and firewall programs)
8) Strange computer behavior
9) Emails/messages being sent automatically and without user's knowledge (a friend receives a strange email from you that you did not send)

III. DATASET

There are many different mobile malware dataset that can be used in the detection of mobile malware. The dataset includes both benign and malicious mobile application samples. Some of these samples used to train detection model and the other samples used for testing. Sherlock, Kaggle, Android Application Dataset, Android Botnet and Drebin are used as a mobile malware datasets.

For example the dataset of [6] is organized as follows: MalGenome dataset used as a public dataset. Their private dataset consists of 30 latest malware from 14 types of malware family. Benign samples include the top 20 free mobile applications that Google Play Store picked. The dataset of [7] includes malicious samples as well as benign samples. Malware samples include 500 malware applications extracted from Contagio Community and 1260 malicious applications from MalGenome. Benign samples contain about 20000 benign applications which extracted randomly from Google Play Store.

The dataset of [8] includes both malicious and benign samples. Malicious samples include 1523 mobile malware applications from DREBIN Dataset while benign sample include 1,709 benign applications from Google Play store.

The dataset of [9] includes both malicious and benign samples. Malicious samples contain 2925 samples from McAfee's internal repository while benign sample contain 3938 samples from McAfee's internal repository.

IV. MOBILE MALWARE ANALYSIS Techniques

Static, dynamic, and hybrid analysis are the three different approaches used for android malware analysis and detection.

A. *Static Analysis*

Static analysis can be defined as a fast and inexpensive approach used for mobile malware detection. This approach can detect mobile malware by examining only the program without executing any code of the program. However static analysis is simple to implement, it produces less information so it limiting the possible features extraction from mobile malware activities.

In [10] the authors presented an android malware detection approach based on examining permission requests that were by android applications. Their model was able to achieve a classification accuracy rate of 80% on their dataset. The dataset came from "Andrototal.org" which contains about 2,444 benign and 870 malicious applications.

In [11] the authors presented a machine learning approach for the detection of android malware based on permissions and APIs. They used a two feature sets, binary and numerical sets. Binary feature set contains 104 binary permission features and 654 binary API features. Numerical feature set contains 104 numerical permission features and 654 numerical API features. Their Android malware applications are from Android MalGenome Project while the benign samples are about 5,000 applications from 25 categories downloaded from Google play.

In [12] the authors presented a machine learning approach for the detection of malware based on dynamic features including resource usage (Memory Usage and CPU Usage) and system calls. These features are collected while the execution of the application. Their dataset consists of 1523 malware applications from Drebin dataset and 1,709 benign applications downloaded from Google Play. In [13] the authors proposed a lightweight approach called RoughDroid for the detection of android malware based on seven feature sets from android manifest file and three feature sets from Dex file. Hardware components, Software components, Requested permissions, Application components, Application Activities, Intent Filters and Application services are the features extracted from manifest file while Suspicious API calls, Restricted API calls and access to undocumented and Hidden APIs are the features extracted from dex file. Their dataset includes Drebin dataset plus 158 Android applications which presenting three new malware families (Grabos,TrojanDropper.Agent.BKY, and AsiaHitGroup) that invade Google Play Store at 2017.

In [14] the authors proposed a feature-based static analysis system called DroidMat for the detection of Android malware. This system detects malware by using various features analyzed and extracted from android manifest file including intent filters to build their malware detection method, the authors used and tested many machine learning classification such as k-means, k-nearest neighbors, and naive Bayes. KNN classifier produces the best accuracy so, it is used in DroidMat to classify application as benign or malware. In [15] The authors proposed a system called HinDroid for detecting android malware based on API calls and an analysis of the relationship between the API calls extracted. Their feature set includes the 200 extracted API calls along with the three different kinds of relationships generated among these API calls (R1, R2, R3). Their dataset consists of Two Datasets from Comodo Cloud Security Center. The first dataset sample set consists of recent collected Android applications from January 30, 2017 to February 5, 2017. This set includes 920 training Android benign application and 914 malware application. This set contains also 500 testing Android samples (198 of them are labeled as good and 302 of them are labeled as malware). The second dataset has a larger collection of android sample containing 30,000 Android applications that were obtained in January 2017, half of which are benign applications and half are malicious. In [16] the authors proposed an efficient APT malware C&C domain detection approach capable of handling unmarked data. In their proposed anomaly detection algorithm, information entropy is introduced to indicate the differing influence of each feature. Their dataset consists of over 300,000 DNS requests from a mobile station every day for two weeks. The dataset contains four feature sets of domain name which are DNS request and answer-based features, Domain-based features, Time-based features and whois-based features.

B. *Dynamic Analysis*

Dynamic Analysis approach detects malware by executing the program and observing the results. This analysis approach detects malware based on mobile malware behavior execution.

In [17] The authors proposed a dynamic android malware detection framework , called EnDroid. EnDroid detects malware from android applications based on multiple types of dynamic behavioral features. These dynamic behavior methods include system-level behavior trace and common application-level malware behaviors like stealing personal information, premium service subscription, and communication of malicious service. EnDroid framework adopts feature selection method chi-square in order to remove noisy or irrelevant dynamic features and for extracting critical dynamic behavior features. Their dataset consists of two dataset; M1 dataset consists of 8806 android benign applications and 5213 malware applications while M2 dataset consists of 5000 benign

applications and 5000 malicious applications. EnDroid applies Stacking and is evaluated on two datasets with various machine learning classifiers (Linear SVM, Naïve Bayes, KNN, Decision Tree, Boosted Tree, Extra Trees, Random Forest and Xgboost) and different feature selection methods. According to Their experimental results, Stacking achieved the best classification performance (96.49% Accuracy, 96.81% precision, 93.65% TPR, 1.82% FPR, 95.91% AUC and 95.21% F-measure for dataset M1) (97.19% Accuracy, 95.25% precision, 97.61% TPR, 1.67% FPR, 97.97% AUC and 96.42% F-measure for dataset M2).

In [18] the authors presented a dynamic analysis and machine learning-based approach for the detection of Android malware by using system calls and network traffic features. They trained their system calls and network traffic classifiers by using 32 malware families of known Android malware families and some typical benign applications. For testing, they used other applications that used for training. They used 120 test application which includes 70 benign applications and 50 malicious applications. Their experimental results stated and indicated that they could achieve 94.02% detection accuracy with J48 and Random Forest machine learning classifier. In [19] the authors proposed 'Dynodroid' which is a dynamic analysis-based system for the detection of Android malware based on the analysis of user interaction. They collected user's activities like long pressing, dragging and tapping the screen. Dynodroid is evaluated on 50 open source Android applications. The authors compared Dynodroid with two prevalent approaches: Monkey and users manually exercising applications. Dynodroid, Monkey and humans covered 55%, 53% and 60% on average, respectively, of each Android applications' Java source code. Dynodroid took 20X less events on average than Monkey. Dynodroid found bugs in Android applications. It found 9 bugs in 7 of the 50 evaluated Android applications, and 6 bugs in the top 5 of 1,000 free Android applications on Google Play Store.

C. Hybrid Analysis

Hybrid analysis forms the combination of both static and dynamic analysis approaches. For improving the detection process of malware, researchers sometimes prefer to apply hybrid analysis which combines both static and dynamic analysis capabilities.

In [20] the authors resented a hybrid approach based on both static and dynamic analysis for the detection of android malware. They used dynamic analysis in order to collect the runtime system calls data of android applications and static analysis in order to analyze the collected system calls data for the detection of malware at a powerful detection server. For identifying and detecting an unknown android application, they use dynamic analysis method in order to collect its system calling data and then compare them with both the malware and benign collected pattern sets offline in order to classify the unknown application. Their dataset consists of a total of 2000 different malware and benign android applications belonging to different categories such as learning, download, tools, games etc. Their detection accuracy exceeds 90%. In [21] the authors proposed ANDRUBIS which is a fully automated large scale and publicly available analysis system for Android applications. This system classifies Android application as benign or malicious based on hybrid analysis. It combines static analysis along with dynamic analysis on both Dalvik VM and system level in order to detect Android malware. It combines also several stimulation techniques for increasing code coverage. Their dataset consists of over one million Android applications, including 40% as Android malicious applications. ANDRUBIS became a publicly available service for the past two years and it accepts public submissions through both web interface and a mobile application. ANDRUBIS is currently capable of analyzing about 3,500 new Android samples per day.

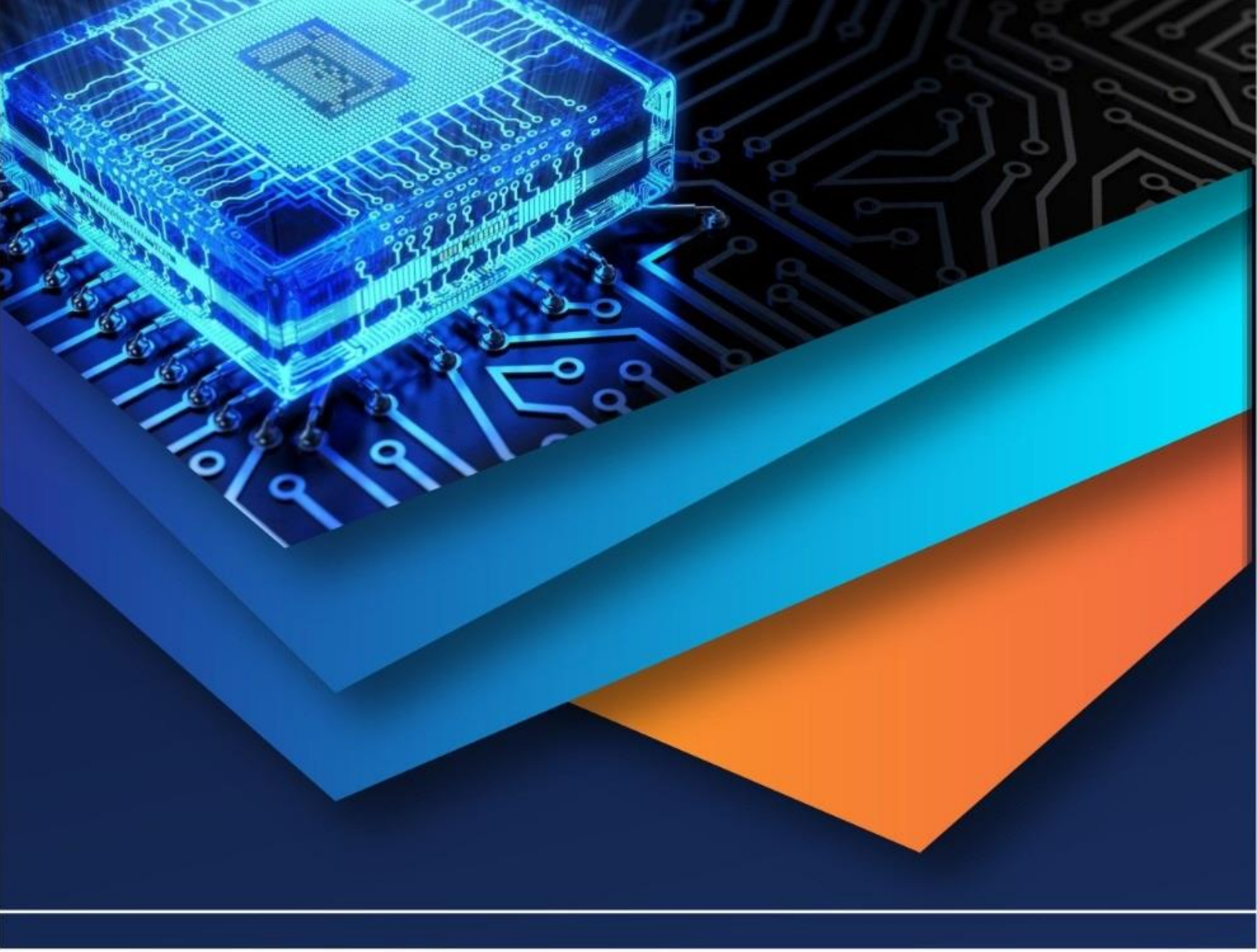
In [22] the authors proposed 'ProfileDroid' which is a hybrid analysis-based multi-layer system for monitoring and profiling Android applications. ProfileDroid published in 2012 in which the authors analyzed and examined Android Manifest.xml (permissions and intents) and java code as static features for static analysis at static layer of the system. User interaction, system calls and network traffic are also considered and analyzed as dynamic features for dynamic analysis at dynamic layer of the system. The authors evaluated ProfileDroid by using Twenty seven free and paid Android applications and they found that 22 applications out of 27 evaluated Android applications communicate with Google during execution.

V. CONCLUSION

This research illustrates mobile malware problem and malware detection analysis approaches. In this study, we discuss mobile malware types, mobile malware datasets and mobile malware analysis techniques. There are three major analysis approaches for the detection of malware. These three major approaches are static, dynamic and hybrid analysis approaches. Static analysis is a fast and inexpensive analysis approach that can be used for mobile malware detection process. It examines a mobile program with no need to execute any code of the program as it can detect mobile malware before the program execution under inspection. Dynamic analysis detects mobile malware after or during the program execution under inspection. Hybrid analysis combines both static and dynamic analysis approaches.

REFERENCES

[1] Malhotra, A. and K. Bajaj, A survey on various malware detection techniques on mobile platform. Int J Comput Appl, 2016. 139(5): p. 15-20.
[2] Salah, A., E. Shalabi, and W. Khedr, A Lightweight Android Malware Classifier Using Novel Feature Selection Methods. Symmetry, 2020. 12(5): p. 858.
[3] Afonso, V.M., et al., Identifying Android malware using dynamically obtained features. Journal of Computer Virology and Hacking Techniques, 2015. 11(1): p. 9-17.
[4] Zulkifli, A., et al. Android Malware Detection Based on Network Traffic Using Decision Tree Algorithm. in International Conference on Soft Computing and Data Mining. 2018. Springer.
[5] Mercaldo, F., et al. Mobile malware detection in the real world. in Proceedings of the 38th International Conference on Software Engineering Companion. 2016. ACM.
[6] Narudin, F.A., et al., Evaluation of machine learning classifiers for mobile malware detection. Soft Computing, 2016. 20(1): p. 343-357.
[7] Yuan, Z., Y. Lu, and Y. Xue, Droiddetector: android malware characterization and detection using deep learning. Tsinghua Science and Technology, 2016. 21(1): p. 114-123.
[8] Ferrante, A., et al. Spotting the malicious moment: Characterizing malware behavior using dynamic features. in Availability, Reliability and Security (ARES), 2016 11th International Conference on. 2016. IEEE.
[9] Yerima, S.Y., S. Sezer, and I. Muttik, High accuracy android malware detection using ensemble learning. IET Information Security, 2015. 9(6): p. 313-320.
[10] Leeds, M. and T. Atkison. Preliminary Results of Applying Machine Learning Algorithms to Android Malware Detection. in Computational Science and Computational Intelligence (CSCI), 2016 International Conference on. 2016. IEEE.
[11] Qiao, M., A.H. Sung, and Q. Liu. Merging permission and api features for android malware detection. in 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI). 2016. IEEE.
[12] Tang, A., S. Sethumadhavan, and S.J. Stolfo. Unsupervised anomaly-based malware detection using hardware features. in International Workshop on Recent Advances in Intrusion Detection. 2014. Springer.
[13] Riad, K. and L. Ke, Roughdroid: Operative scheme for functional android malware detection. Security and Communication Networks, 2018. 2018.
[14] Wu, D.-J., et al. Droidmat: Android malware detection through manifest and api calls tracing. in 2012 Seventh Asia Joint Conference on Information Security. 2012. IEEE.
[15] Hou, S., et al. Hindroid: An intelligent android malware detection system based on structured heterogeneous information network. in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017. ACM.
[16] Niu, W., et al., Identifying APT Malware Domain Based on Mobile DNS Logging. Mathematical Problems in Engineering, 2017. 2017.
[17] Feng, P., et al., A Novel Dynamic Android Malware Detection System With Ensemble Learning. IEEE Access, 2018. 6: p. 30996-31011.
[18] Su, X., M. Chuah, and G. Tan. Smartphone dual defense protection framework: Detecting malicious applications in android markets. in 2012 8th International Conference on Mobile Ad-hoc and Sensor Networks (MSN). 2012. IEEE.
[19] Machiry, A., R. Tahiliani, and M. Naik. Dynodroid: An input generation system for android apps. in Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering. 2013.
[20] Tong, F. and Z. Yan, A hybrid approach of mobile malware detection in Android. Journal of Parallel and Distributed computing, 2017. 103: p. 22-31.
[21] Lindorfer, M., et al. Andrubis--1,000,000 apps later: A view on current Android malware behaviors. in 2014 Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS). 2014. IEEE.
[22] Wei, X., et al. ProfileDroid: multi-layer profiling of android applications. in Proceedings of the 18th annual international conference on Mobile computing and networking. 2012. ACM.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY