IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 4, April 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304427 |

Malicious Mobile Applications Identification and Analysis System

Nusrath, Priya M M, Priya Shree M, Shreelekha M.K , Yogaraja G S R, Ms Eshwaridevi Jagapur

U.G. Student, Department of Information Science and Engineering, S J C Institute of Technology, Chickballapur, Karnataka, India

U.G. Student, Department of Information Science and Engineering, S J C Institute of Technology, Chickballapur, Karnataka, India

U.G. Student, Department of Information Science and Engineering, S J C Institute of Technology, Chickballapur, Karnataka, India

U.G. Student, Department of Information Science and Engineering, S J C Institute of Technology, Chickballapur, Karnataka, India

Assistant Professor, Department of Information Science and Engineering, S J C Institute of Technology, Chickballapur, Karnataka, India

Assistant Professor, School of Computer Science and Engineering, REVA University, Bangalore, India

ABSTRACT: The popular mobile operating system, Android, is based mostly on an open-source Linux kernel and a few other programs. Since its initial release in September 2008, this operating system has drawn the attention of malevolent developers. There are currently more than 2.3 billion gadgets. Consequently, in order to correctly identify harmful 11111 applications, it is imperative that Android applications be studied and analyzed. Two main techniques for analyzing Android applications to distinguish between malicious and benign ones are static and dynamic assessments. This article describes a study that uses a variety of machine learning models to examine many Android applications. By the end of the year, malware on the Android operating system platform had significantly increased—by 400%. This is due to assaults on11111 Android applications, which are sometimes shortened to "Apps," are simpler than their desktop equivalents. Malware can be introduced in a variety of ways, and most of the time, consumers are unaware that the malicious feature is there.

Users of Android frequently provide rights to Android apps without giving it much attention. Android's essential security is preserved through methods called permissions. Paradoxically, if applications are given unauthorized permissions, they may be able to access private data and acquire dangerous powers. Enter your desired changes in this section. Then, use the button below to paraphrase. It really is that simple!

KEYWORDS: Malware, Static Analysis, Dynamic Analysis, Malicious Applications, Identification

I. INTRODUCTION

These rights can be used by malicious apps to reveal private data about the user, including phone contacts, gallery photographs, credit card details, and more. Occasionally, once the user begins using it, the malware or harmful behavior becomes apparent. Finding and properly analyzing malware in Android apps is hence significant and appropriate. This project focuses on the analysis of Android applications in two modes: dynamic mode, which records the behavior of the application in runtime by utilizing multiple machine learning models, and static mode, which involves reverse engineering the application and verifying the XML file. The project's main goal is to investigate how well various machine learning models analyze and identify Android malware. In order to gauge their level of safety, the research also examines a number of the most popular Android apps in Bangladesh.

Bouncer, an internal malware detection system, is used by Google Play Store. However, researchers found that this system's capacity to detect malware was far from ideal. When an application is flagged as harmful by the Google Play Store, it is possible that the malware has already caused enough harm to the device system by the time it is discovered. Furthermore, there exist unofficial app stores from which malware cannot be detected with any underlying inspection. Installed apps from these marketplaces may readily include dangerous code. It's difficult to find such harmful code. Nonetheless, studies have demonstrated that the application of machine learning algorithms to identify fraudulent activity frequently yields extremely high accuracy. The subsequent subsections offer a concise synopsis of the primary analytical methodologies employed, namely static and dynamic analysis, together with a selection of relevant literature for each respective category.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304427 |

II. LITERATURE REVIEW

The most popular operating system for smartphones is now android. When compared to previous years, the quantity of malwares has significantly increased due to the rapidly growing use of Android. Numerous anti-malware applications are available that are made to effectively defend user sensitive data on mobile platforms from these Kinds of attacks. .. Here, I've looked at the many Android malwares and their deep learning-based attack techniques, as well as antivirus software that protects Android systems from malware. Next, we talked about many deep learning- based methods for detecting malware on Android, including DroidDeepLearner, DroidDeepDetector, Deep Flow, Droid Delver, and Droid Deep. Our goal is to put into practice a deep learning model that can determine, without installation, whether an Android application is infested with malware or not. Edit sorts, requires a significant quantity of tagged image data in order to be prepared.

Poland's WarsawElayan, Omar N. et al. The most popular operating system for smartphones is now android. In comparison to previous years, the quantity of malwares has significantly increased due to the rapidly expanding use of Android. Numerous antimalware applications are available that are made to effectively defend user sensitive data on mobile platforms from these kinds of attacks. Here, I've looked at the many Android malwares and their deep learning-based attack techniques, as well as antivirus software that protects Android systems from malware. The several deep learning-based methods for detecting Android malware that we have discussed include DeepFlow, Droid Delver, Droid Deep, Maldozer, DroidDetector, and DroidDeepLearner. Our goal is to develop a deep learning- based model that can detect, without the need for installation, whether an Android application is tainted with malware.

One Nor Badrul Anuar and Ahmad Firdaus A crucial component of internet-connected device security is malware detection. Dynamic malware analysis makes advantage of several characteristic combinations. The various combinations are produced via Summary Information, DLLs, which Registry Key Changes, and APIs. The dynamic malware analysis tool, Cuckoo Sandbox, offers good accuracy and may be customized. Using PEFILE, 92 features are statically extracted from binary malware and more than 2300 features are extracted from dynamic malware analysis. Static features are extracted from 10,000 benign files and 39000 dangerous binaries. In Cuckoo Sandbox, 2200 malicious files and 800 benign files are dynamically examined, and 2300 features are extracted. While static analysis yields an accuracy of 99.36%, dynamic malware analysis yields an accuracy of 94.64%. Because malware has clever and cunning behaviors, dynamic malware analysis is ineffective. Because network activity is controlled, dynamic analysis has some constraints and cannot be fully studied because of restricted network access.

Ankita Kapratwar The rising popularity of Android mobile devices has led to an increase in the prevalence of Android malware in recent years. Android malware poses serious threats to consumers, including fraud and the revealing of personal information, and is installed on mobile devices without the user's permission. Additionally, some malware may infect the mobile device and inflict harm by hiding inside it utilizing different obfuscation tactics. The researcher suggested a range of detecting methods to lessen these hazards. Regretfully, it's still difficult to identify Android spyware. keeps getting bigger. Currently, malware is detected using signature-based detection; however, new and undiscovered malware cannot be identified using this method. This indicates that, given the vast disparity in current research and techniques, creating an efficient virus detection system is essential.A deep learning program using multispectral images is proposed by Xiaoxiao Liu, Yutong Li, Yue Lv, Xiaochen Guo, Yang Gao, and Wen Zhang to predict the germination potential of maize seeds. They achieved a 92.8% accuracy rate. The main drawback of this approach is that it necessitates a multispectral imaging system, which may not be feasible in environments with limited resources.Michael Spreitzenbarth2, Hugo Gascon1, Daniel Arp1, and Malte Hubner Android is one of the most popular smartphone operating systems available today. With hundreds of thousands of applications in different industries, it provides its users with an abundance of capabilities. Regrettably, malicious spyware is being installed on Android smartphones at a rapid pace by cybercriminals. In contrast to other platforms, Android allows apps from untrusted sources, such as third-party shops, to be installed, which makes it simpler for attackers to bundle and distribute malware-filled programs. Over 55,000 dangerous programs and 119 new malware families were discovered in 2012 alone, according to a recent investigation.It is clear that action must be taken to stop malware from spreading throughout Android markets and mobile devices.Sharma Akanksha and Dash Subrat Kumar The rise in popularity of mobile devices has radically and quickly changed how people use computers and their routines around them. The most widely used mobile operating system, Android, uses a complex permission control mechanism to provide a privilege-separated security system. Permissions are required for Android apps to access sensitive personal data and system resources, however actual research has shown that malicious software can get permissions to attack systems and apps by tricking users and circumventing security measures. In this work, we suggest a unique machine learning method for identifying

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304427 |

malware by analyzing the permissions and API function calls patterns that Android apps obtain and utilize.. For both qualitative and quantitative assessment, binary and numerical features are extracted using static analysis of the source code and resource files of Android apps. To decrease the feature dimension and increase efficiency, feature selection techniques are used.Jiquiang Lu and Xing Lu In recent years, malware developers have turned their attention to the Android platform. A permission-based access control mechanism is one of Android's primary defenses against rogue apps. Using the permissions an application sought to identify a potentially dangerous application is a workable method. In order to identify malicious Android applications, we suggested a two-layered permission-based detection method in this research. In contrast to earlier studies, we take into account the apps' requested permission pairs as an extra condition, and we also take into account the permissions that are actually used to increase detection accuracy. Based on an evaluation with 28548 benign apps and 1536 malicious apps, the evaluation's findings show that a two layered permission-based detector has a high malware detection rate.

III. EXISTING SYSTEM

The rapid increase in malware has focused researchers' attention on malware detection methods. The tools that developers and the Android community use to examine applications for malware are known as malware detectors. These detectors' effectiveness and quality are evaluated based on specific methods they use. There are three main types of invasive detection approaches that were used: host- based, cloud-based, and social collaboration. Static, dynamic, and hybrid classifications are applied to all detection techniques. For anomaly-based or signature-based techniques, the strategy depends on how the data to be utilized for malware identification is gathered. Systems that employ anomaly-based malware detection must go through a training phase in order to identify the system's typical behavior and to make comparisons.

In this instance, the malware detection system is educated. However, there are drawbacks to this approach as well. Didier Stevens' experiment made it abundantly evident that if a malware script is zero padded with a specific number of zero bytes, it gets challenging and occasionally stays undiscovered by his virus scanners. Furthermore, a website with more than 60 anti-virus scanners combines a variety of antivirus programs and internet search engines to decide if a file submitted to the website is contaminated or not. It even identifies viruses that the user's antivirus program missed or reported as false positives.
An further disadvantage of the signature-based technique is that the source code can be altered and reorganized to prevent the occurrence of common signatures, which makes it challenging for the antivirus program to gather and produce a signature for that particular sample. The process of producing a signature gets challenging even for viruses that are polymorphic or metamorphic. This is because the body of polymorphic viruses varies each time they are moved to a new system since a decrypter/encrypter stub is employed along with a new key.However, in the case of metamorphic viruses, the virus has the ability to successfully recode itself and adapt.

IV. PROPOSED SYSTEM

Objectives of the Proposed Work
Learn about different malware and how it affects an Android device. Additionally, about the various machine learning methods and their variations, as well as about the permissions that Android applications require. Discover the many virus types and how they impact Android devices. Furthermore, regarding the many machine learning techniques and their variants, and concerning the permissions that Android applications need. Android sample data is converted to pictures. And applying CNN to the pictures and dividing them into dangerous and harmless categories. Outcomes of the Proposed System

Enhanced Mobile App Security: With real-time detection and analysis of harmful apps, MMAIAS is intended to greatly enhance Mobile app security. Users don't have to worry about unintentionally downloading dangerous programs to utilize their mobile devices with confidence. Prompt Threat Detection: Quick threat identification is made possible by the system's real-time scanning and analyzing features.Through prompt identification, the risk of harmful applications causing data breaches, financial losses, or unauthorized access is reduced. Decreased Risk Exposure: By using the threat assessments that MMAIAS provides, users and businesses can decide which applications to install or trust, with more knowledge.The total risk exposure related to using mobile apps is lowered as a result.MMAIAS aids in user education by warning users about possible dangers and outlining the reasons for an app's classification as dangerous. More caution and security-conscious conduct may result from this raised awareness. Enhanced Incident Response:By incorporating MMAIAS into existing incident response procedures, businesses can get better results.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

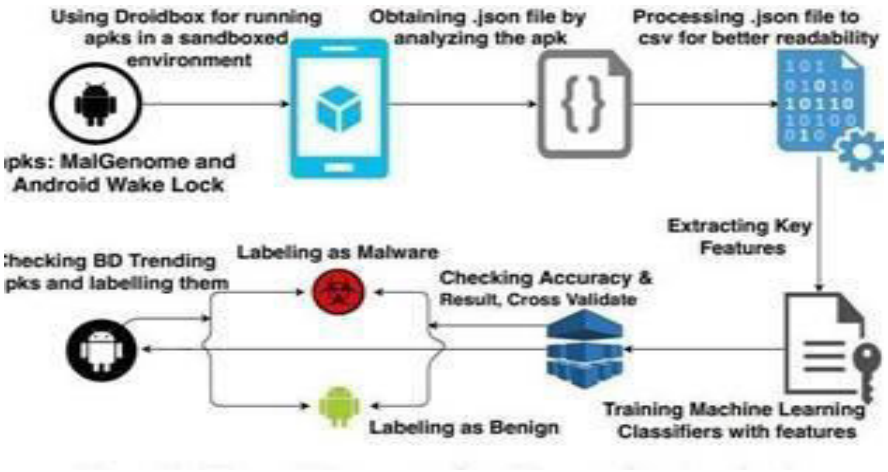
|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304427 |

Administrators may remove dangerous apps from the market right away by removing their rights or placing impacted devices in quarantine.Customization of Security Policies: Administrators may tailor MMAIAS security policies to meet their own security needs and preferences.

V. METHOLODOGY

In order to forecast seed germination and assess grain quality for various crops, the methodological schematic combines IoT devices with deep learning algorithms. The primary phases include data gathering utilizing Internet of Things (IoT) tools to monitor ecological conditions, CNN- based model construction for more accurate germination prediction, real-time control, and proactive management through warnings and suggestions.
Using an easy-to-use interface that provides decision guidance, farmers will utilize a visual depiction of the forecasts. To improve the functionality and performance of this system in agriculture, validation and ongoing research are essential. Block Diagram



In order to solve these problems, first a feature set was created by extracting, processing, and vectoring heterogeneous knowledge.Second, a normalcy model that explains the classifier's typical behavior is used to train a K-NN classifier. This makes it possible to identify anomalous behavior by looking for actions that differ from the listed typical actions of programs. The use of novelty classifiers, which only require one class, is an additional strategy that can be used. In this instance, the trained system just needs to complete one class to learn with typical system activity, allowing the anomalous behavior to be inferred. Any machine learning technique utilized, regardless of how sophisticated or sophisticated the machine algorithm is, can never be fully efficient.

1. **Malware Sample Identification:** The majority of anti- virus scanners currently in use employ the signature-based technique, as was previously discussed in the section on classical techniques. However, many viruses evade detection by modern anti-virus programs and are only detected when the system is severely compromised, approximately three to four months after they first enter it. Therefore, it is evident that a quicker and more efficient solution to the aforementioned issues is required in order to slow down the spread o f malware within the Android system.

2. **Using Java based Android package analyzer:** To lessen the likelihood that malware in the Android community would avoid being discovered by scanners, we are using an approach in this project that aims to identify both known malware families and undiscovered malware. In order to do this, we assembled a dataset from numerous android .apk samples that we got from Virus Share, Google Play, and other reliable websites that offer malware samples.

3. **Using Python based Android package analyzer :** However, this was a laborious procedure that required a very long time to extract permissions for a large amount of data. As a result, we wrote a Python script that can read and process several samples at once, access the manifest.xml file, extract permissions, and combine those permissions into a CSV file that can be used by machine learning methods. The Python script was executed on the identical

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304427 |

samples numerous times in order to guarantee accurate data and reduce the likelihood of randomization. The finished product is error-free, stored, and used as a dataset in our project.

4. **Using machine learning:** supervised learning algorithms. The classifier is then trained using the previously created dataset and then tested to predict the result according to the feature vectors. In using machine learning, we encountered two main problems, at first the need to extract some variety of feature illustration of the applications and second, the requirement for a knowledge set that's virtually completely benign or well labelled which is able to be wont to train a classifier.

In order to solve these problems, first a feature set was created by extracting, processing, and vectoring heterogeneous knowledge.Second, a normalcy model that explains the classifier's typical behavior is used to train a K-NN classifier. This makes it possible to identify anomalous behavior by looking for actions that differ from the listed typical actions of programs. The use of novelty classifiers, which only require one class, is an additional strategy that can be used. In this instance, the trained system just needs to complete one class to learn with typical system activity, allowing the anomalous behavior to be inferred. Any machine learning technique utilized, regardless of how sophisticated or sophisticated the machine algorithm is, can never be fully efficient.

VI. CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, by utilizing machine learning approaches for efficient analysis and detection, this article seeks to solve the urgent problem of malware threats in the Android ecosystem. We have illustrated the possibility of improving Android security using automated and intelligent methods by thoroughly investigating static and dynamic assessments. The static analysis model, in contrast, showed an impressive 81% accuracy.

These findings highlight the need of adding dynamic analysis to the detection process in order to supplement the knowledge obtained from static analysis. In addition, the examination of popular Indian apps yielded insightful contextual information that clarified regional security issues and patterns.

This geographic component deepens our comprehension of the Android security environment. This project's success rests not only in detecting malware with high accuracy rates but also in the useful application of user-friendly interfaces and automated response mechanisms. Proactive defensive approach is enhanced by the automated response system, which guarantees prompt action against identified threats.

We looked at the two most widely used techniques for analyzing malware: static and dynamic analysis. Both were carried out in great detail, and a dataset was produced for dynamic analysis, which allowed us to identify malware and benign apps with up to 93% accuracy. Cross- Validation was also done in order to strengthen the study's conclusion.

Our research leads us to the conclusion that dynamic analysis may, in fact, be more effective than static analysis at identifying Android malware. The accuracy of dynamic analysis was significantly higher than that of static analysis, and it may be possible to improve upon this finding by incorporating new features.

Future Enhancements: This research will go forward into hybrid analysis, which combines static and dynamic analysis techniques. We believe hybrid analysis can produce even greater outcomes. In order to improve performance, we also want to develop a Neural Network model for the analysis. In addition to developing an open- source platform where users may upload and test programs to determine whether or not they are
malicious, the authors hope to carry out further in-depth study on the analysis of Android applications and malware.

REFERENCES

[1] "Android Malware Detection Using Deep Learning" The 2nd International Workshop on Data-Driven Security (DDSW2021)March 23 - 26, 2021, Warsaw, PolandOmar N. Elayan et al. / Procedia Computer Science 184 (2021) 847–852.
[2] "Root-exploit Malware Detection using Static Analysis and Machine Learning" 1, 2Ahmad Firdaus, 1Nor Badrul Anuar The 4th International Conference on Computer Science and Computational Mathematics (ICCSCM 2015) .
[3] Kapratwar, Ankita, "Static and Dynamic Analysis for Android Malware Detection" (2016).Master's Projects

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304427 |

SJSUScholar Works.

[4] "DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket"Daniel Arp1, Michael Spreitzenbarth2, Malte H¨ubner1, Hugo Gascon1, Konrad Rieck,2002.

[5] "Mining API Calls and Permissions for Android Malware Detection" Akanksha Sharma & Subrat Kumar Dash, "20th international confererence,2002.

[6] "A Two-Layered Permission-Based Android Malware Detection Scheme" Xing Lu,Jiquiang Lu ,Technical report IDE1202,February 201. D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective andexplainable detection of android malware in your pocket." inNdss, vol. 14, 2014, pp. 23–26.

[7] Sharma and S. K. Dash, "Mining api calls and permissions for android malware detection," in International Conference on Cryptology and Network Security. Springer, 2014, pp. 191–205.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details