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.2020.6046

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

Malware Detection & Prevention in Android Mobile by using Significant Permission Identification & Machine Learning

Ms. Kirti Reddy[1], Mr. Danesh Bastani[2], Ms. Charul Joshi[3], Mr. Abhijeet singh[4], Prof. Rinku Badgujar[5]
[1, 2, 3, 5]Computer Engineering, [4]Assistant Professor, BSIOTR, Pune, India

Abstract: ***Malware is today one of the biggest security threat to internet. People today use terms such as malware, spyware, or ransomware much more than the term "virus" where Malware can steal your information, make your device send SMS messages to premium rate text services, or install adware that forces you to view web pages or Download apps.***
We need a robust malware detection solution to counter this serious malware project that can effectively and efficiently recognis e malware apps. We will present SIGPID in our proposed system to perform data extraction and as SIGPID design is effective in malware detection. Our method defines the substantial work permission needed by an application and differentiate between essential and non-essential permissions and detect and remove the malware on such a basis.
Keywords: ***Machine learning, classification, malware, android, SIGPID, SVM***

I. INTRODUCTION

The malware infection epidemic has been so serious that a new study reveals that 97 percent of all Android apps contain mobile malware. During 2016, more than 3.25 million new fake Android devices have been found. This turns into an appearance of a new malicious Android device every approximately every 10 seconds. These malicious apps are created to execute various types of attacks in the form of trojan , worms, hacks, and viruses. Many notorious malicious apps have over 50 variants which makes it incredibly difficult to locate them all. Developers and researchers have employed diverse strategies to develop malware identification tools for Android to resolve these increasing protection issues.

RISKRANKER, Uses static review, for example, to detect harmful behavior in Android applications. However, set methods to analysis typically presume that more acts are expected than will currently be necessary, which may lead to a variety of false positives. To boost the precision of the tests, researchers have proposed different complex analysis approaches to catch the real-time execution context. TAINTDROID, for example, utilizes tainting analysis to monitor many important data points at the same time. However, dynamic research methods typically allow sufficient input suites to effectively exercise paths of execution. Since we can use cloud computing to meet these requirements, the privacy issues then become a potential issue. Petra et al . show that there is a wide range of anti-analysis techniques that advanced malware can use to successfully avoid dynamic analytics-based malware detection.More recent attempts to analyze consumer behavioral data in both the android platform and other online data collection applications have been made. The required privileges from the applications were used to implement the least privilege, which to some degree shows the functionalities of the apps as well as the act ions of runtime.

As a result, researchers use techniques of machine learning and data mining to detect Android malware based on use of permission. For example, DREBIN[6] uses multi-view technology to combine static analysis with supervised learning to detect malware accurately. SIGPID [7] strengthens DREBIN[6] through the use of many more learning and detection functions. Droidclassifier[8] utilizes traffic flow knowledge and unsupervised learning to detect malware and classify the kin of the malicious software. In this paper , we present SIGPID, an approach that draws substantial permissions from apps and uses the extracted data to effectively detect malware using supervised learning algorithms. The aim of SIGPID 's development is to detect malware efficiently and reliably. As stated earlier, the number of newly introduced malware increases without precedent. It would encourage analysts to become more actively involved in the detection and analysis of malware as such. Our method analyzes permissions and only defines permissions that distinguish between malicious and benign applications. In specific, they propose a multi-level data pruning approach for strategically harvesting negative rating permission rankings, association-based permission mining and support-based permission ranking substantial permissions. Then, classification algorithms based on machine learning are used to identify various types of malware and benign applications.

Our empirical research findings indicate that when using Vector Machine Help (SVM) as a classifier, SIGPID will substantially reduce the amount of permissions we need to test to only 22 out of 135 (84 percent reduction), thus maintaining more than 90 percent

accuracy in malware detection and Fmeasure. We also see that our approach lists smaller numbers of important permits than the number of "dangerous" permissions that Google has identified. Only eight authorisations with their list appear on our list. As a data-driven method, SIGPID defines meaningful allowances dynamically based on actual applications and not on the static concept of dangerous permissions based on their intended services.

This fundamental difference helps us find more malware using the dangerous list alone than the method. We also check SIGPID with 67 widely used supervised algorithms to show the generality of this method, and consider that with all these algorithms it retains very high precision.

Furthermore, we contrast the accuracy and runtime quality of our approach with two state-of-the-art methods, DREBIN, Permission-Induced malware detection and current virus scanners..

We also find that our approach can detect more Malware samples than the other approaches with considerably less overhead. In brief, our paper contributes the following:

A. We are developing SIGPID, an approach that recognizes a significant subset of permissions that can be used effectively to detect Android malware. The amount of allowances to be calculated would be decreased by 84 percent using our approach.

B. We assess the success of our approach with just one fifth of the total amount of App permissions. We believe that SIGPID can achieve more than 90 % accuracy, reminder, reliability and F-measurement. These findings are in favor of a strategy that uses all 135 permissions and a dangerous list of permissions. We will consider that SIGPID is more efficient when detecting 93.62% of maliciously apps in the data package and 91.40% of unknown malware relative to other state-of-the-art malware detection approaches..

C. We use SIGPID with 67 widely used supervised learning algorithms and a much larger dataset (5,494 malicious and 310,926 benign apps) to demonstrate that a wide range of supervised learning algorithms can generally work with the technique. We assume that 55 out of 67 algorithms can reach F-measurement of at least 85%, while the average runtime can be decreased by 85.6% relative to the baseline system.

II. LITERATURE SURVEY

A. IDC, "Smartphone OS market share, 2017 q1." [Online].Available:https://www.idc.com/promo/smartphon market-share/OS:

Mobile phones used to control tiny portable devices which are increasingly growing and have become the practical component of humans. The new era's openness has contributed to domain demand and implementatio by allowing strong association of electronic assistance such as digital banking. Competition under developers is in terms of power, processors, capacity, usability functionality, camera, etc. but the key driver for the competitive demand and success is the operating system of smartphones. It work addresses operating system pros and cons and analyzes participant success over others by recommending board improvement focuses to make them more appropriate and safe.

B. M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, "Riskranker: scalable and accurate zero-day android malware detection," in Proceedings of the 10th international conference on Mobile systems, applications and services. ACM, 2012, pp. 281–294.

Recent explosive growth in smart-phone sales has occurred. The success also promotes the deployment of malware authors using malicious programs (or applications) into various mobile markets. These dangerous applications hide themselves in a vast range of other innocuous devices that are hard to identify. Existing smartphone antivirus program is not adequate to use recognized malware samples to collect indicators despite its constructive existence. In this study , author had proposed a proactive scheme for Android to spot zero-day malware.

Researchers are inspired, using malware samples and their signatures, to determine the possible security danger presented by such ineffective services. The authors developed an automatic Riskranker system for scalable analysis of hazardous activities (e.g., root exploit release or SMS background sending). The output is then used to establish a priority list of growing applications requiring further analysis. Applied to 118, 318 applications downloaded in September and October 2011 from different Android markets, the researcher program needs less than four days to process them all, and effectively records 3, 281 dangerous results. Of such recorded applications, 718 samples of malware (in 29 families) have been effectively discovered, and 322 are zero-day (in 11 families). This results demonstrate Riskranker 's effectiveness and scalability for Android policing in any way.

C. S.Wang, Q.Yan, Z.Chen, B.Yang, C.Zhao, and M.Conti,"Textdroid: Semantics-based detection of mobile malware using network flows."

Android is the most popular mobile operating system market share of 80 per cent, but as a result it is often the most malware-

targeted device. Malware researchers typically use analytics tools to automate the extraction of device features to counter the rising number of wild malicious Android applications. While the research community has addressed the significance of such instruments, the resulting prototypes keep their analytical capacity and availability constrained. In this study the author proposed ANDRUBIS, a fully automated, open-to-public and detailed evaluation program for Android apps. ANDRUBIS combines static analysis and dynamic analysis at the Dalvik VM and system level, as well as growing the scope of codes by stimulus techniques. We've got over 1,000,000 ANDRUBIS Android applications including a malware base of 40 percent. This dataset helps us to speak about malware trends seen in apps from 2010, and to gain insight into how ANDRUBIS has been used over the last 2 years as a publicly accessible resource.

D. *J. Z. L. H. P. S. Y. Lichao Sun, Xiaokai Wei and W. Srisa-an, "Contaminant removal for android malware detection systems," in Proceedings of IEEE International Conference on Big Data, 2017*

Threats are prone to the digitalization period, it becomes more troublesome because it may escape by copying the anti-threat habits of innocuous software by implementing the same functionality as transmitting tweets, then squashing the payload to reduce the chance of getting trapped by invoking payload at night. Their protection responsive tool analyzes, which are used as networks, are on the maps. Indirect approaches of viruses such as this distinguish the vulnerable and harmless aspects of software by in-depth research. Its idea is focused on manipulating factors used to breach innocuous and malicious applications' security-related behavior. Basically, security-sensitive behavior based on an app context is extracted using a static analysis program to compare the nature of benign and malware within applications. Implementation of a prototype is proposed in this research by evaluating app-context from malware dataset on 202 malicious apps, and 633 existing benign apps on play store. 192 malicious apps are recognized via app context with 95 per cent recall and 87.7 per cent precision. Research indicates that behavioral intent is directly linked to security-sensitive information maliciousness (reflected by actions) rather than form of security-sensitive sources obtained through behavior.

E. *X. Cao, L. Liu, W. Shen, A. Laha, J. Tang, and Y. Cheng, "Real-time misbehavior detection and mitigation in cyber-physical systems over wlans," IEEE Transactions on Industrial Informatics, vol. 13, no. 1, pp. 186–197, 2017*

Mass level vulnerability may be expected to increase quickly following the spike in mobile network demands at present. Network interface is used as a portal for fraudulent acts, such as stealing user's personal details and conducting destructive practices on an entity or agency. Within this work they suggest Text-droid, integrating natural language processing with machine learning makes processes of threat analysis efficient and automatic. Malware samples are identified by extracting recognizable features by Text-droid. Aid vector machine classifier is used to build model for detecting mobile threats utilizing malware. Advance SVM models reflect powerful output across two key data sets, Test system embracing a 96.36 percent threat detection rate and an device collection occupying 76.99 percent in the wild, Flow Header Visualization method is built in advance to visualize large text produced through network interactions with apps, Apps dynamic activities are evaluated by protection researchers.

F. *S. Wang, Q. Yan, Z. Chen, B. Yang, C. Zhao, and M. Conti, "Detecting android malware leveraging text semantics of network flows," IEEE Transactions on Information Forensics and Security, 2017.*

A recent study says a new malicious software is released every 4 seconds This fast distribution of malware enables existing malware detection systems to fall far behind, encourages malicious apps to bypass monitoring efforts and even official app stores to spread a range of negative effects as trustworthy network malware circulates. Second, the proliferation of these malicious apps could easily and widely cause devices to get infected. In addition , researchers and authorities depending on machine-based identification techniques could often deploy and mistakenly label these programs as harmless, because they were not identified because malware. Such tests are then used as part of their benign dataset to process the training and testing. Contaminants in benign data may affect the effectiveness and accuracy of their detection and identification techniques. To solve this issue, author proposed PUDROID (Positive and Unlabeled Learning-Based Android Malware Detection) for automated and effective reduction of malware classification and detectors by computer to apprentice. Allowing machine learning to identify and diagnose malware in a more effective and reliable way. The researcher uses a selection technique to choose appropriate features from a range of features to further enhance the efficiency of these detector systems. Researchers then test detection levels and precision of the detection method using two databases, with another PUDROID being added without pollutants for contaminant detection. The findings indicate that the Author can greatly increase rates of malware identification as well as precision when extracting pollutants from the datasets.

G. *T. Cruz, L. Rosa, J. Proenca, L. A. Maglaras, M. Aubigny, L.Lev,J. Jiang, and P. Simoes, "A cybersecurity detection framework for supervisory control and data acquisition on Industrial Informatics, vol. 12, no. 6, pp. 2236–2246, 2016.*

By deliberately handling its protocol parameters in cyber-physical systems (CPS) over IEEE 802.11e-based local wireless zone networks (WLANs), a deceptive node can gain significant advantage over other ordinary nodes in terms of resource sharing. Owing to the unpredictable exposure to the protocol, it is challenging to classify the misbehaving node accurately even in real time. Several new misbehaviors, built specifically for common IEEE802.11e networks and heterogonal network configurations, are inapplicable. In this work , the researcher suggested range of novel countermeasures for real-time and low-weight use, including a hybrid-share detector for misuse and a CPS packet-dropping device based on IEEE 802.11e. A develop mathematical models for the proposed detector 's effectiveness and mitigation mechanisms. Extensive simulation findings demonstrate that the methods suggested will achieve a high detection rate and punish a malfunctioning node with a high fall rate.

III. CURRENT SYSTEM

Android is actually the most commonly adopted network holding 85 percent of the market share with multiple android consumers downloading applications from specific sources and pages offering malicious software. According to a survey on malware infection, that consists of various malware present over 97 percent of devices were targeted. Some researchers have developed various approaches to tackling such a problem.

Taintdroid: This program uses complex methods to monitor critical data by analytical tainting

Riskranker: This system uses static approach to detect malicious application

A. Limitations of Current System

1) Static analyzes used in risk ranker can result mainly in false positives.
2) Technical research used to conduct execution of taint robot require adequate feedback Then privacy security is issue.
3) Only malware is detected, not deleted.
4) Vulnerable to malware with ease.

IV. PROPOSED SYSTEM

In our proposed system we are going to present SIGPID to perform extraction of data and as design of SIGPID is effective in detection of malware. Our approach identifies the significant permission required by an application to work. And differentiate between essential and non-essential permissions and on such basis the malware is detected and removal is performed.

A. System Architecture

Figure shows the system architecture of the model proposed in our work. The model comprises of following components – Installation fil e, APK Tool, XML Parser, Extracted Permissions, Static and Dynamic analysis, Feature extraction, binary feature vector, 3 layer net and RNN.

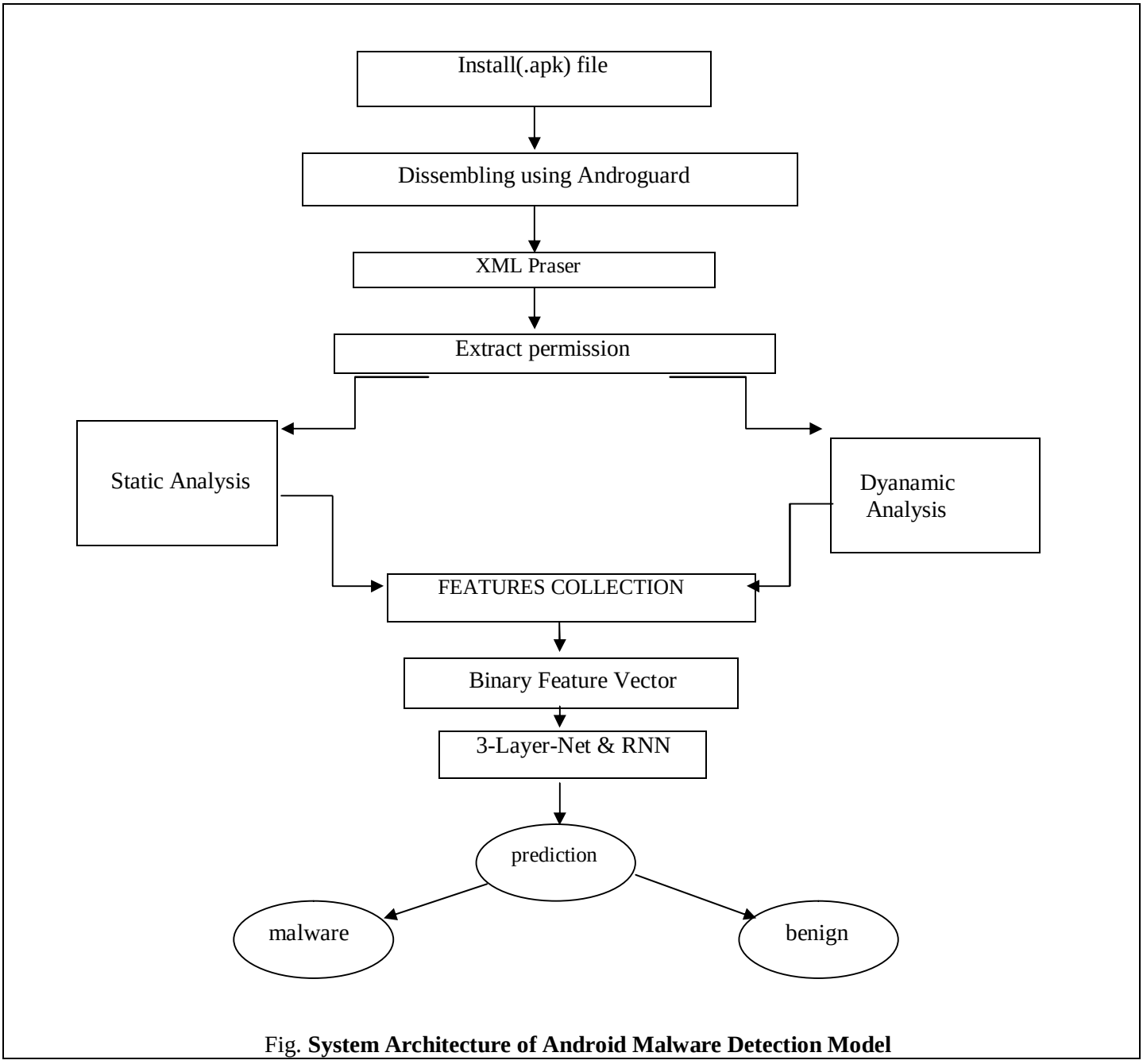


Fig. **System Architecture of Android Malware Detection Model**

B. Feature Extraction

We collected total of 500 malicious applications and 20,000 benign applications from Contagio community. We performed static and dynamic analysis on the applications collected in order to extract the features for each of the application.

1) *Types*: required permissions, sensitive APIs, and dynamic behaviours. Among them, required permissions and sensitive APIs are extracted through the static analysis, whereas dynamic behaviours are extracted through dynamic analysis. An important static analysis technique focuses on analysing the manifest file (AndroidManifest.XML) included in every application for the set of permissions used and other components like Services, Broadcast receivers and Intents. Dynamic analysis includes extracting the dynamic behaviour of app that are included when the app is running. All we need is the installation file i.e.,.apk file for each of the app that is collected from dataset. We them uncompressed the .apk file.

As a first step each application is disassembled using Apktool to generate the manifest file and the source code. Then the permissions from the manifest file are extracted using an XML parser written in Python. We parsed these two files AndroidManifest.xml and classes.dex file. We thus obtain the permissions required by the app using Androguard decompiling tool. Androguard is a full python tool to analyse Android files; dex, odex, APK, Android's binary xml, Android resources, disassemble dex byte codes and decompiler for dex files. To extract features from any android app, we just need to use some quick command line commands that will print app permissions as output on the screen. For example, we can use the get_permissions() command like follows:

```
Command: get_permissions()
In [19]: a.get_permissions()

Out[19]:['android.permission.RECEIVE_BOOT_COMPLETED','android.permission.INT
ERNET','android.permission.READ_PHONE_STATE','android.permission.WRITE_EXTER
NAL_STORAGE','android.permission.ACCESS_NETWORK_STATE','android.permission.S
END_SMS','android.permission.RECEIVE_SMS']
```

Fig 2: get_permissions() command as used in Androguard for extracting permissions

V. CONCLUSION

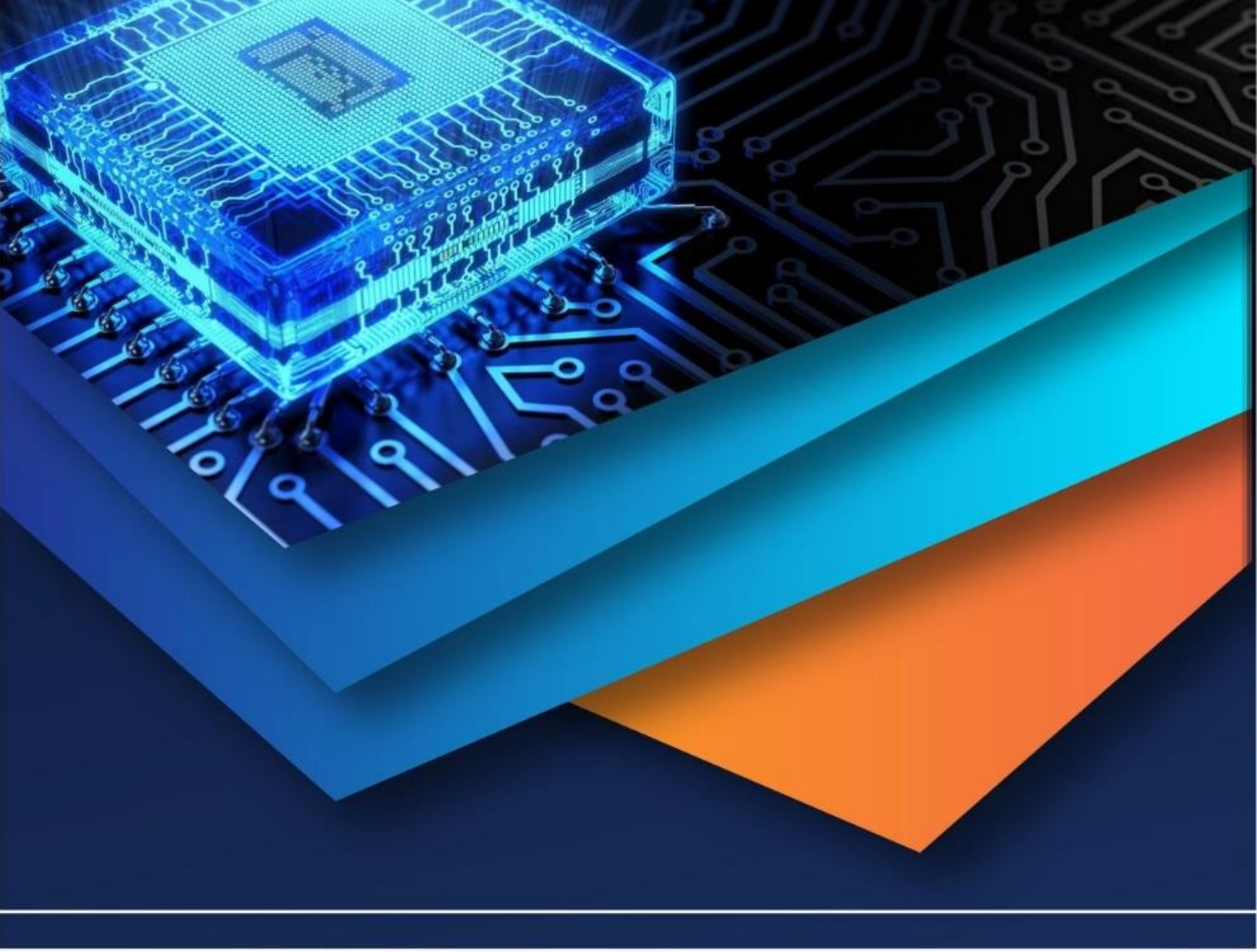
Existing systems have been examining the mechanism for classifying Android apps, whether they are malicious or ordinary. So the main focus of the existing system is on detecting malicious apps, here we are preventing the malicious app by uninstalling those apps that have been detected. The malware detection system survey therefore analyzes malware applications based on the permission list using SIGPID, and analyzes malware prevention

VI.ACKNOWLEDGMENT

The All faith and respect for her goodness and inspiration to our HOD. I want to thank all my Friends and Family members for their care of us. Huge gratitude to our, Project Leader, Project Guide and all other staff members for supplying us with input on this paper.

REFERENCES

[1] IDC, “Smartphone os market share, 2017 q1.” [Online].Available:https://www.idc.com/promo/smartphone-market- Share/os .
[2] M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, “Risk ranker: scalable and accurate zero-day android malware detection,” in Proceedings of the 10th international conference on Mobile systems, applications, and services. ACM, 2012, pp. 281–294.
[3] S. Wang, Q. Yan, Z. Chen, B. Yang, C. Zhao,and M. Conti, “Textdroid: Semantics-based detection of mobile malware using network flows,” in IEEE INFOCOM 2017.
[4] J. Z. L. H. P. S. Y. Lichao Sun, Xiaokai Wei and W. Srisa-an, “Contaminant removal for android malware Detection systems,” in Proceedings of IEEE International Conference on Big Data, 2017.
[5] X. Cao, L. Liu, W. Shen, A. Laha, J. Tang, and Y. Cheng, “Real-time misbehavior detection and mitigation in cyber-physical systems over wlans,” IEEE Transactions on industrial informatics,vol 13, no. 1,pp. 186-197, 2017.
[6] S. Wang, Q. Yan, Z. Chen, B. Yang, C. Zhao, and M. Conti, “Detecting android malware leveraging text semantics of network flows,” IEEE Transactions on Information Forensics and Security, 2017.
[7] T. Cruz, L. Rosa, J. Proenca, L. A. Maglaras, M. Aubigny, L.Lev,J. Jiang, and P. Simoes, “A cybersecurity detection framework for supervisory control and data acquisition on Industrial Informatics, vol. 12, no. 6, pp. 2236–2246,2016.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY