Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, No. 4, May : 2023

MALWARE DETECTION IN ANDROID USING MACHINE LEARNING

Mrs. REVATHI K Head of the Department, Information Technology, SNS College of Engineering, Coimbatore, Tamil Nadu621107, India, revathi.k.it@snsce.ac.in
Mr.PAWANA PRUDHVI T S V Department of Information Technology, SNS College of Engineering, Coimbatore, TamilNadu 621107: pawanprudhvi88@gmail.com
Mr.RANJITHKUMAR Department of Information Technology, SNS College of Engineering, Coimbatore, TamilNadu 621107, India, thisisranjiths@gmail.com
Mr. SUGANESHWARAN .E Department of Information Technology, SNS College of Engineering, Coimbatore, TamilNadu 621107, India, suganeshwaran63@gmail.com
Mr. ARUN KUMAR R Department of Information Technology, SNS College of Engineering, Coimbatore,TamilNadu 621107, India, arunabi1918@gmail.com

Abstract:
With the market share of Android system becoming the first in the world, the security problem of Android system is becoming more and more serious. How to effectively detect Android malware has become a significant problem. Permissions and API calls in Android applications can effectively reflect the behaviour patterns of an Android application. Most researchers have only considered a single permission or API feature, and did not consider associations and patterns inside the permission or API features. Some scholars have also tried to find the combination modes inside the permission features in malwares, but the detection of maliciousness according to this combination mode is too absolute. This paper proposes a malware detection method, which combines the advantages of frequent pattern mining and Naive Bayes to effectively identifyAndroid malwares.

Keywords: Classification, cybersecurity, API..

I INTRODUCTION:

In this technological era, smartphone usage and its associated applications are rapidly increasing due to the convenience and efficiency in various applications and the growing improvement in the hardware and software on smart devices. It is predicted that there will be 4.3 billion smartphone users by 2023 . Android is the most widely used mobile operating system (OS). As of May 2021, its market share was 72.2% . The second highest market share of 26.99% is owned by Apple iOS, while the rest of the 0.81% is shared among Samsung, KaiOS, and other small vendors . Google Play is the official app store for Android-based devices. The number of apps published on it was over 2.9 million as of May 2021. Of these, more than 2.5 million apps are classified as regular apps, while 0.4 million apps are classified as low-quality apps by AppBrain .

Android's worldwide popularity makes it a more attractive target for cybercriminals and is more at risk from malware and viruses. Studies have proposed various methods of detecting these attacks, and ML is one of the most prominent techniques among them . This is because ML techniques are able to derive a classifier from a (limited) set of training examples. The use of examples thus avoids the need to explicitly define signatures in developing malware detectors. Defining signatures requires expertise and tedious human involvement and for some attack scenarios explicit rules (signatures) do not exist, but examples can be obtained easily.

Numerous industrial and academic research has been carried out on ML-based malware detection on Android, which is the focus of this review paper. Android users and developers are known to make mistakesthat expose them to unnecessary dangers and risks of infecting their devices with malware.

II. RELATED WORKS

However, ML-based machine learning methods have not been thoroughly reviewed as the main focus is onlyon the static analysis techniques.

In , a survey was carried out using existing literature up to 2017 to identify malware detection techniques together with their advantages and disadvantages. Under static and dynamic analysis, they have grouped several approaches that can be used to identify Android malware. However, the analysis of this survey was not comprehensive as it focused on a limited number of studies. Based on the previous studies, a systematic review was conducted in . According to it, there are five types of Android malware detection techniques. They are static detection, dynamic detection, hybrid detection, permission-based detection, and emulation-based detection. They also summarized the reviewed work with the model accuracy of malware detection, but the approach of those studies was not discussed. The review conducted in analyzed several studies conducted until 2019 related to ML models which can be used to detect Android malware. The malware and APK analysis methods were not discussed in detail since the focus on identifying different ML models was the priority in this review. It is better to analyses the accuracies of the identified ML models. The novel ML/DL and other models which can be used to detect Android malware were also not in the focus of this review. The review in provides a good analysis of static, dynamic, and hybrid detection techniques used in the existing research studies for Android malware detection. Along with that possibility of using machine learning models, several deep learning models are also discussed. However, this study did not comprehensively analyze the model accuracy of the machine learning methods for Android malware detection since this study focused more on discussing different malware detection approaches instead of considering the accuracy of those approaches. Hence, these works differ from our study.

In , a systematic review on DL-based methods for Android malware defense was discussed. Malware detection ,malware family detection, repackaged/fake app detection, adversarial learning attacks and protections, and malicious behavior analysis were identified as the malware defines objectives in this review together with the usage of DL models. Though they have identified the possible DL models, it is still better to analyses the accuracy and compare it with traditional ML methods and other hybrid approaches.

III. PROPOSED SYSTEM

IMPLEMENTATION OF DESIGN THINKING CONCEPT

Design thinking is a problem-solving approach that is focused on understanding the needs and desires of users in order to create innovative and effective solutions. It is a process that encourages creativity, experimentation, and iteration, and can be applied to a wide range of design challenges.

The design thinking process typically involves the following stages:

1. Empathy
2. Define
3. Ideate
4. Prototype
5. Testing

PROBLEM STATEMENT

The increasing prevalence of Android devices and the rapid growth of mobile applications have led to an increased threat of malware attacks on these devices. Malware can cause serious damage to users' personal information, device functionality, and privacy. Traditional signature-based methods of detecting malware on Android devices have limitations, as they rely on identifying known malware signatures and may not be effective in detecting new and emerging malware threats. Additionally, malware authors can use various obfuscation techniques to evade signature-based detection.To overcome these limitations, new approaches to malware detection on Android devices are needed. The proposed method aims to address this need by using a combination of static and dynamic analysis techniques to provide comprehensive and accurate detection results. The proposed

method involves four main stages: permission analysis, code analysis, behavior analysis, and user feedback. By combining these stages, the proposed method is able to identify both known and emerging malware threats.The proposed method is expected to provide an effective solution to the problem of detecting malware on Android devices. By using multiple analysis techniques, the proposed method can detect malware that may not be detected by traditional signature-based methods. Additionally, the proposed method can provide users with a report detailing the findings of the analysis, allowing them to make an informed decision about whether to uninstall the application or take other action. The proposed method has the potential to enhance the security of Android devices and protect users from malware attacks.

EMPATHY

- Malware detection in Android using machine learning is an important research area, as it can help to improve the accuracy and efficiency of malware detection. Empathy is a machine learning-based approach to Android malware detection that has been proposed in recent research.
- Empathy is a system that uses a deep learning-based approach to detect Android malware. The system extracts a set of features from the Android application, including permissions, API calls, and resource files, and feeds them into a deep neural network for classification.
- Nowadays, the usage of mobile devices or smartphones has increasing in our daily and almost all of the people around world own a smartphone. According to Global market share, during second quarter of 2018, there was 88% smartphones in the market have been sold towards end users and that is Android systems

DEFINE

- Malware detection in Android using machine learning refers to the use of machine learning techniques to identify and classify malicious software, or malware, that targets the Android mobile operating system.
- Some common machine learning algorithms used in Android malware detection include decision trees, random forests, support vector machines, and neural networks. These algorithms can be trained on a large dataset of labeled Android applications, where the labels indicate whether an application is benign or malicious. Once trained, the algorithms can then be used to classify new, unlabeled applications as either benign or malicious.

IDEATE

- AndroidManifest.xml file uses to get essential information like permission and activities of application information. All the permission that has been extracted needs to be saved as x.arff file and loaded in WEKA. Meanwhile, the permission values stored as binary number (0 or 1).
- Moreover, the optimization feature is used to help in gaining the best features of permissions. The permission features of malware detection were trained and classified by using significant features.
- This study applied the features selection to get an outstanding feature for the best malware detection. Using Chat GPT, we can generate text and relevant quality images and videos automatically by analyzing patterns in large datasets of human written text. This allows us to create realistic and coherent content on a wide range of topics.

IV TECHNOLOGIES USED:

JavaServer Pages (JSP) is a Java technology that allows software developers to dynamically generate HTML, XML or other types of documents in response to a Web client request. The technology allows Java code and certain pre-defined actions to be embedded into static content.

JSPs are compiled into Java Servlets by a JSP compiler. A JSP compiler may generate a servlet in Java code that is then compiled by the Java compiler, or it may generate byte code for the servlet

directly. JSPs can also be interpreted on-the-fly reducing the time taken to reload changesJavaServer Pages (JSP) technology provides a simplified, fast way to create dynamic web content. JSP technology enables rapid development of web-based applications that are server- and platform-independent.

Android Emulator -A virtual mobile device that runs on our computer -use to design, debug, and test ourapplications in an actual Android run-time environment

Android Development Tools Plugin -for the Eclipse IDE - adds powerful extensions to the Eclipse integrated environment

Dalvik Debug Monitor Service (DDMS) -Integrated with Dalvik -this tool let us manage processes on anemulator and assists in debugging

Android Asset Packaging Tool (AAPT) – Constructs the distributable Android package files (.apk)

Android Debug Bridge (ADB) – provides link to a running emulator. Can copy files to emulator, install .apkfiles and run commands.

The Java Servlet API allows a software developer to add dynamic content to a Web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML. Servlets are the Java counterpart to non-Java dynamic Web content technologies such as PHP, CGI and ASP.NET. Servlets can maintain state across many server transactions by using HTTP cookies, sessionvariables or URL rewriting.

The Servlet API, contained in the Java package hierarchy javax.servlet, defines the expected interactions of a Web container and a servlet. A Web container is essentially the component of a Web server that interacts with the servlets. The Web container is responsible for managing the lifecycle of servlets, mapping a URL to. a particular servlet and ensuring that the URL requester has the correct access rights

V . WORKING

1. Data Collection: The first step is to collect a dataset of Android applications that includes both benign and malicious applications. The dataset should be diverse and representative of the types of applications that arecommonly used on Android devices.
2. Feature Extraction: The next step is to extract features from the Android applications that can be used to train the machine learning model. Some common features used in Android malware detection include permissions,API calls, resource files, and system calls.
3. Data Preprocessing: The extracted features are then preprocessed to remove noise and prepare them for use in the machine learning model. This may involve scaling, normalization, and other techniques to ensure that thedata is suitable for the machine learning algorithms.
4. Training the Model: The preprocessed data is then used to train the machine learning model. Various algorithms such as decision trees, random forests, support vector machines, and neural networks can be usedto train the model.

APPROACHES IN DETECTING MALWARE

Bottom Up Testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the topof the hierarchy is tested.

All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage.

Top Down Testing is an approach to integrated testing where the top integrated modules are tested and thebranch of the module is tested step by step until the end of the related module.

Sandwich Testing is an approach to combine top down testing with bottom up testing.
The main advantage of the Bottom-Up approach is that bugs are more easily found. With Top-Down, it iseasier to find a missing branch link
SEO Optimization: ChatGPT can be programmed to generate content that is optimized for search engine rankings, which can help businesses improve their online visibility and attract more traffic to their website.
Verification and Validation are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it full fills its intended purpose. These are critical components of a quality management system such as ISO 9000. The words "verification" and "validation" are sometimes preceded with "Independent" (or IV&V), indicating that the verification and validation is to be performed by a disinterested third party.
It is sometimes said that validation can be expressed by the query "Are you building the right thing?" and verification by "Are you building it right?"In practice, the usage of these terms varies. Sometimes they areeven used interchangeably.
The PMBOK guide, an IEEE standard, defines them as follows in its 4th edition "Validation. The assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers. Contrast with verification."
"Verification. The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process. Contrast with validation." Verification is intended to check that a product, service, or system (or portion thereof, or set thereof) meets a set of initial design specifications. In the development phase, verification procedures involve performing special tests to model or simulate a portion, or the entirety, of a product, service or system, then performing a review or analysis of the modelling results. Malware for Android is becoming increasingly dangerous to the safety of mobile devices and the data they hold. Although machine learning techniques have been shown to be effective at detecting malware for Android, a comprehensive analysis of the methods used is required. We review the current state of Android malware detection using machine learning in this paper. We begin by providing an overview of Android malware and the security issues it causes. Then, we look at the various supervised, unsupervised, and deep learning machine learning approaches that have been utilized for Android malware detection. Additionally, we present a comparison of the performance of various Android malware detection methods and talk about the performance evaluation metrics that are utilized to evaluate their efficacy. Finally, we draw attention to the drawbacks and difficulties of the methods that are currently in use and suggest possible future directions for research in this area. In addition to providing insights into the current state of Android malware detection using machine learning, our review provides a comprehensive overview of the subject.

VI CONCLUSION

Any smartphone is potentially vulnerable to security breaches, but Android devices are more lucrative for attackers. This is due to its open-source nature and the larger market share compared to other operating systems for mobile devices. This paper discussed the Android architecture and its security model, as well as potential threat vectors for the Android operating system. Based upon the available literature, a systematic
review of the state-of-the-art ML-based Android malware detection techniques was carried out, covering the latest research from 2016 to 2021. It discussed the available ML and DL models and their performance in Android malware detection, code and APK analysis methods, feature analysis and extraction methods and strengths and limitations of the proposed methods, Malware aside, if a developer makes a mistake, it is easier for a hacker to find and exploit these vulnerabilities. Therefore, methods for the detection of source code vulnerabilities using ML were discussed. The work identified the potential gaps in previous research and possible future research directions to

enhance the security of Android OS.

Both Android malware and its detection techniques are evolving. Therefore, we believe that similar future reviews are necessary to cover these emerging threats and their detection methods. As per our findings in this paper, since DL methods have proven to be more accurate than traditional ML models, it will be beneficial to the research community if more comprehensive systematic reviews can be performed by focusing only on DL based malware detection on Android. The possibility of using reinforcement learning to identify source code vulnerabilities is another area of interest in which systematic reviews and studies can be carried out.

VII FUTURE ENHANCEMENT

Malware for Android is becoming increasingly dangerous to the safety of mobile devices and the data they hold. Although machine learning techniques have been shown to be effective at detecting malware for Android, a comprehensive analysis of the methods used is required. We review the current state of Android malware detection using machine learning in this paper. We begin by providing an overview of Android malware and the security issues it causes. Then, we look at the various supervised, unsupervised, and deep learning machine learning approaches that have been utilized for Android malware detection. Additionally, we present a comparison of the performance of various Android malware detection methods and talk about the performance evaluation metrics that are utilized to evaluate their efficacy. Finally, we draw attention to the drawbacks and difficulties of the methods that are currently in use and suggest possible future directions for research in this area. In addition to providing insights into the current state of Android malware detection using machine learning, our review provides a comprehensive overview of the subject.

VIII REFERENCES

- Number of Mobile Phone Users Worldwide from 2016 to 2023 (In Billions). Available online: http s:// www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/(accessedon 19 May 2021).
- Mobile Operating System Market Share Worldwide. Available online: https://gs.statcount er.com/so-market-share/mobile/worldwide/ (accessed on 19 May2021).
- Available online: https://www.appbrai n.com/ stats/number-of-android-apps/ (accessed on 19 May 2021).
- Gilbert, D.; Mateo, C.; Planes, J. The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. J. Newt. Compute. Appl. 2020, 153, 102526. [Google Scholar][Crossruff]
- Khan, J.; Shahzad, S. Android Architecture and Related Security Risks. Asian J. Technol. Manga. Res. [ISSN: 2249-0892] 2015, 5, 14-18. Available online: http://www.ajtmr.com/papers/V ol5Issue2/Vol5lss2_P4.pdf (accessed on 19 May 2021).
- Platform Architecture. Available online: https://developer.android.com/guide/platform (accessedon 19 May 2021).