Vol 5, Issue 2, 2025

Malware Detection A Framework for Reverse Engineered Android Applications through Machine Learning Algorithms

Mrs K. Indumathi[1],N. Shirisha[2],T. Udaya Sri[3]

[1]Assistant professor, Department of CSE, Princeton College of engineering and technology for women Narapally vijayapuri colony ghatkesar mandal, Pin code-500088

[2,3]UG Students, Department of CSE, Princeton College of engineering and technology for women Narapally vijayapuri colony ghatkesar mandal, Pin code-500088

ABSTRACT

Today, Android is one of the most used operating systems in smartphone technology. This is the main reason, Android has become the favorite target for hackers and attackers. Malicious codes are being embedded in Android applications in such a sophisticated manner that detecting and identifying an application as a malware has become the toughest job for security providers. In terms of ingenuity and cognition, Android malware has progressed to the point where they're more impervious to conventional detection techniques. Approaches based on machine learning have emerged as a much more effective way to tackle the intricacy and originality of developing Android threats. They function by first identifying current patterns of malware activity and then using this information to distinguish between identified threats and unidentified threats with unknown behavior. This research paper uses Reverse Engineered Android applications' features and Machine Learning algorithms to find vulnerabilities present in Smartphone applications. Our contribution is twofold. Firstly, we propose a model that incorporates more innovative static feature sets with the largest current datasets of malware samples than conventional methods. Secondly, we have used ensemble learning with machine learning algorithms such as AdaBoost, SVM, etc. to improve our model's performance. Our experimental results and findings exhibit 96.24% accuracy to detect extracted malware from Android applications, with a 0.3 False Positive Rate (FPR). The proposed model incorporates ignored detrimental features such as permissions, intents, API calls, and so on, trained by feeding a solitary arbitrary feature, extracted by reverse engineering as an input to the machine.

Index Terms—-Malware Detection, Machine Learning.

I.INTRODUCTION

1.MOTIVATION

The project stems from the escalating threat of malicious Android applications that compromise user security and privacy. With the rapid evolution of sophisticated malware, traditional detection methods prove insufficient. Recognizing the value of reverse engineering in understanding malicious software, this project aims to develop a robust framework integrating machine learning algorithms for the automated analysis of reverse-engineered Android applications. By leveraging machine learning, the project seeks to enhance the accuracy and efficiency of malware detection, contributing to a more secure and resilient mobile ecosystem.

2.PROBLEM DEFINITION

The problem at the core of this endeavor is the detection of Android malware, a challenge exacerbated by the limitations of traditional signature-based methods in keeping pace with evolving threats. Reverse engineering offers a valuable means to comprehend application internals, yet the analysis process is intricate and time-consuming. The project addresses this problem by proposing a framework that automates the analysis of reverse-engineered Android applications using machine learning algorithms. The goal is to develop models capable of discerning between benign and malicious behaviors, identifying novel threats, and establishing a proactive defense against emerging security risks.

3.OBJECTIVE OF PROJECT

The primary objective of the project is the creation of a comprehensive framework for detecting malware in Android applications through the application of machine learning algorithms to reverse-engineered

data. This involves developing automated tools for reverse engineering, implementing machine learning models for distinguishing between benign and malicious patterns, focusing on behavioral analysis to enhance detection capabilities, providing real-time malware detection during application installation or execution, and ensuring adaptability to new and emerging threats through continuous updates and model training.

4.SCOPE OF PROJECT

The scope of the project encompasses several key areas, including in-depth reverse engineering analysis, feature extraction from reverse-engineered data, training of machine learning models, real-time detection integration, scalability to handle diverse applications, and user-friendly design for adoption by security professionals and developers. By addressing these objectives within the defined scope, the project aims to advance the field of Android malware detection, contributing to improved security practices for mobile devices.

II.LITERATURE REVIEW

1.Malware Detection: A Framework for Reverse Engineered Android Applications Through Machine Learning Algorithms, Android has emerged as a dominant operating system in the realm of smartphone technology, attracting the attention of hackers and attackers due to its widespread usage. The sophistication of malware embedded within Android applications presents a formidable challenge for security providers, necessitating innovative approaches for detection and identification. In response to this challenge, machine learning-based methods have emerged as a more effective means of combating evolving Android threats. Leveraging static features extracted from reverse engineered Android applications and employing ensemble learning techniques with machine learning algorithms such as AdaBoost and Support Vector Machine, this research paper proposes a model that achieves high accuracy in detecting malware, exhibiting a 96.24%

accuracy rate with a minimal False Positive Rate (FPR). By incorporating advanced static feature sets and employing ensemble learning, the proposed model demonstrates improved performance in identifying vulnerabilities present in Smartphone applications.

2.Android Malware Classification Using Machine Learning and Bio-Inspired Optimization Algorithms, The proliferation of Android malware in recent years has necessitated the development of robust classification frameworks for identifying malicious applications. This paper introduces a prototype framework that utilizes static analysis methods and two feature sets—permissions declared in the AndroidManifest.xml and Android classes from the Classes.dex file—to classify Android malware. Machine learning algorithms, including Random Forest, SGD, SVM, and Neural Networks, are trained using the extracted features and further optimized using bio-inspired optimization algorithms such as Particle Swarm Optimization, Artificial Bee Colony optimization (ABC), Firefly optimization, and Genetic algorithm. The prototype framework achieves high accuracy rates across different datasets, with notable results including a 99.6% accuracy using an SGD classifier for the Andro-Dump dataset. The paper underscores the importance of feature selection and optimization techniques in enhancing the accuracy and effectiveness of Android malware classification frameworks.

3.Android Malware Permission-Based Multi-Class Classification Using Extremely Randomized Trees, With the increasing reliance on smartphones for various activities, including business, health, and financial transactions, the proliferation of mobile applications has led to a rise in malicious applications targeting users' sensitive information. This research presents a reverse engineering framework (RevEng) that utilizes machine learning algorithms to analyze applications' permissions and classify them based on their malicious intent. By employing extremely randomized trees, the framework achieves high accuracy and reduced execution time by selecting a reduced set of permissions. Two approaches—binary value representation and feature importance—are explored, with both approaches demonstrating improved accuracy and performance compared to existing methods. The paper highlights the efficacy of permission-based multi-

class classification in identifying malicious applications and underscores the importance of feature selection and optimization in enhancing classification accuracy and efficiency.

III.EXISTING SYSTEM AND DIADVANTAGES:

The prevailing approaches to Android malware detection often face limitations. Signature-based methods, commonly used in existing systems, struggle to keep up with the rapid evolution of malware, as they rely on known patterns. Additionally, heuristic-based systems may generate false positives due to their reliance on predefined rules, leading to misclassification of benign apps as malicious. Furthermore, these methods might not effectively handle polymorphic and obfuscated malware, reducing their overall detection accuracy. The manual efforts involved in creating and updating signatures and rules can also impede the scalability of the existing systems, especially in dealing with the increasing volume and diversity of Android malware.

IV.PROPOSED SYSTEM AND ADVANTAGES

The proposed system aims to address the shortcomings of the existing approaches by integrating a novel framework for Android malware detection. The advantages of the proposed system include:

Machine Learning-Based Detection: The incorporation of machine learning algorithms enables the system to adapt and learn from evolving malware patterns, enhancing accuracy and reducing false positives.

Behavioral Analysis: The proposed system focuses on behavioral markers, examining the dynamic behavior of Android applications. This approach enables the detection of previously unknown malware by identifying anomalous behavior based on

permission requests and other behavioral indicators.

Reduced False Positives: Leveraging machine learning for feature extraction and analysis contributes to a more nuanced understanding of application behavior, minimizing false positives compared to traditional heuristic-based systems.

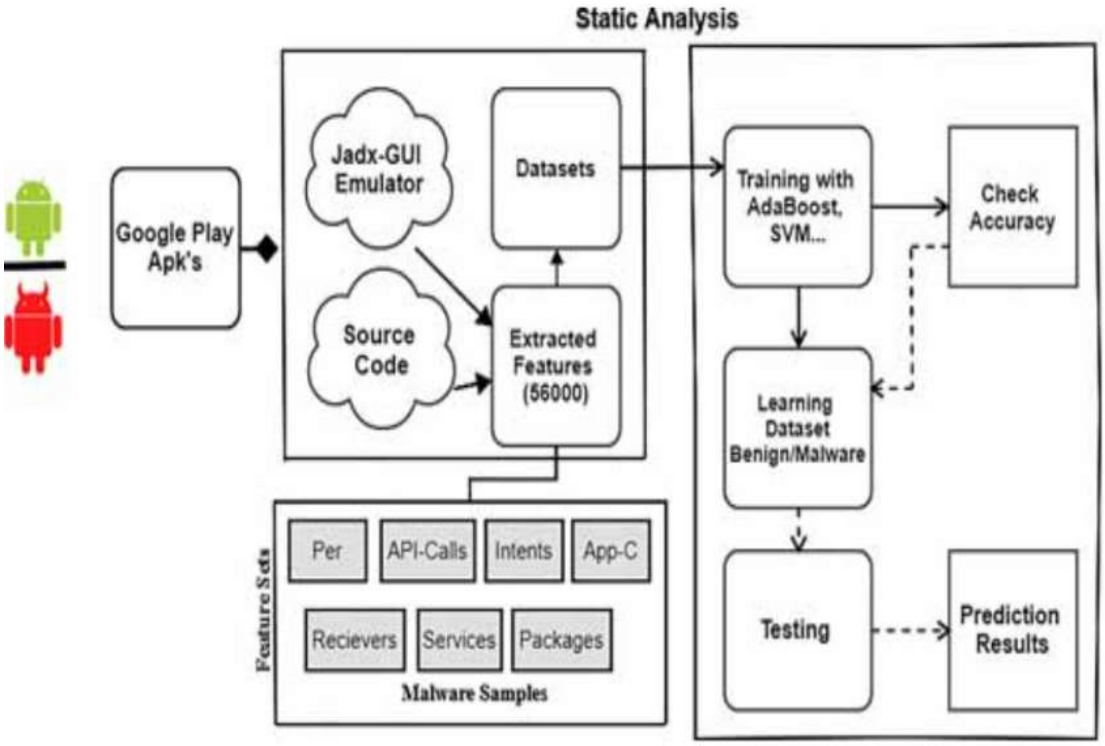

Fig 1. SYSTEM ARCHITECTURE

V.MODULES

- **Data Collection Module:** The Data Collection Module serves as the foundation for the framework by gathering information from reverse-engineered Android applications. It conducts both static and dynamic analyses during application execution, extracting relevant features crucial for subsequent stages. This module ensures that comprehensive data is collected for effective analysis by the system.
- **Feature Extraction Module:** The Feature Extraction Module is

responsible for identifying and extracting pertinent features from the collected data. It analyzes static features such as permissions requested and API calls, along with processing dynamic features like runtime behavior and system interactions. The module converts raw data into feature vectors, preparing the input for the machine learning models.

- **Machine Learning Module:** The Machine Learning Module is at the core of the framework, implementing machine learning algorithms for effective malware detection. It involves training models using labeled datasets, validating their accuracy, and integrating them into the system for real-time detection during application installation or execution. This module is crucial for enhancing the system's ability to discern between benign and malicious behaviors.

- **Real-Time Detection Module:** The Real-Time Detection Module ensures the constant monitoring and analysis of applications during installation or execution. It triggers the machine learning models for on-the-fly detection, comparing application behavior against known malware patterns. This module plays a vital role in providing timely alerts and taking preventive actions upon detecting potential threats.

- **User Interface (UI) Module:** The User Interface Module provides a user-friendly front-end for system interaction. It displays relevant information on detected malware, allowing users to initiate scans or customize detection settings. Clear alerts and recommendations are presented through the interface, ensuring effective communication between the system and users.

- **Database Management Module:** The Database Management Module handles the storage and retrieval of datasets and model parameters. It manages historical data for analysis and model training, ensuring efficient retrieval for real-time detection. This module is crucial for maintaining a well-organized and accessible repository of information.

- **Reporting and Logging Module:** The Reporting and Logging Module generates reports and logs for analysis and user awareness. It records detection outcomes and reasons, providing detailed reports

on detected malware. System administrators and users can access logs to gain insights into the system's activities, contributing to transparency and accountability.

VI.CONCLUSION

In conclusion, the proposed framework for Android malware detection represents a significant advancement in addressing the challenges posed by the existing methods. By leveraging machine learning algorithms and prioritizing behavioral analysis, the system introduces adaptability, learning capabilities, and a proactive defense mechanism against emerging threats. The emphasis on real-time detection during application installation or execution enhances responsiveness, ensuring the timely identification and mitigation of malicious activities.

The scalability of the proposed system, achieved through the reduction of manual efforts in signature creation and rule definition, facilitates the handling of a growing dataset of diverse Android applications. The continuous learning mechanism guarantees adaptability to evolving threats, maintaining a high level of detection accuracy over time. The system's focus on minimizing false positives, aided by nuanced feature extraction and analysis through machine learning, contributes to optimized resource utilization and improved overall performance.

In essence, the proposed framework offers a comprehensive and forward-thinking solution for Android malware detection, striving to elevate the effectiveness, adaptability, and efficiency of the detection process within the dynamic landscape of mobile security. Through these innovations, the project aims to contribute significantly to the enhancement of Android application security, ensuring the safety and privacy of mobile device users in the face of evolving cybersecurity challenges.

VII.REFERENCES

1.Android (GOOG) Just Hit a Record 88% Market Share of All Smartphones—Quartz, Jan. 2022, [online] Available: https://qz.com/826672/android-goog-just-hit-a-record-88-market-shareof-all-smartphones/.

2.A. O. Christiana, B. A. Gyunka and A. Noah, "Android malware detection through machine learning techniques: A review", *Int. J. Online Biomed. Eng.*, vol. 16, no. 2, pp. 14, Feb. 2020.

3.D. Ghimire and J. Lee, "Geometric feature-based facial expression recognition in image sequences using multi-class AdaBoost and support vector machines", *Sensors*, vol. 13, no. 6, pp. 7714-7734, Jun. 2013.

4.R. Wang, "AdaBoost for feature selection classification and its relation with SVM a review", *Phys. Proc.*, vol. 25, pp. 800-807, Jan. 2012.

5.J. Sun, H. Fujita, P. Chen and H. Li, "Dynamic financial distress prediction with concept drift based on time weighting combined with Adaboost support vector machine ensemble", *Knowl.-Based Syst.*, vol. 120, pp. 4-14, Mar. 2017.

6.A. Garg and K. Tai, "Comparison of statistical and machine learning methods in modelling of data with multicollinearity", *Int. J. Model. Identificat. Control*, vol. 18, no. 4, pp. 295, 2013.

7.C. P. Obite, N. P. Olewuezi, G. U. Ugwuanyim and D. C. Bartholomew, "Multicollinearity effect in regression analysis: A feed forward artificial neural network approach", *Asian J. Probab. Statist.*, vol. 6, no. 1, pp. 22-33, Jan. 2020.

8. W. Wang, M. Zhao, Z. Gao, G. Xu, H. Xian, Y. Li, et al., "Constructing features for detecting Android malicious applications: Issues taxonomy and directions", *IEEE Access*, vol. 7, pp. 67602-67631, 2019.

9.B. Rashidi, C. Fung and E. Bertino, "Android malicious application detection using support vector machine and active learning", *Proc. 13th Int. Conf. Netw. Service Manage. (CNSM)*, pp. 1-9, Nov. 2017.

10.J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-An and H. Ye, "Significant permission identification for machine-learning-based Android malware detection", *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3216-3225, Jul. 2018.

11.G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez and J. Blasco, "Dendroid: A text mining approach to analyzing and classifying code structures in Android malware families", *Exp. Syst. Appl.*, vol. 41, no. 4, pp. 1104-1117, Mar. 2014.

12.M. Magdum, "Permission based mobile malware detection system using machine learning", *Techniques*, vol. 14, no. 6, pp. 6170-6174, 2015.

13.M. Qiao, A. H. Sung and Q. Liu, "Merging permission and API features for Android malware detection", *Proc.*

5th IIAI Int. Congr. Adv. Appl. Informat. (IIAI-AAI), pp. 566-571, Jul. 2016.

14.D. O. Sahin, O. E. Kural, S. Akleylek and E. Kilic, "New results on permission based static analysis for Android malware", *Proc. 6th Int. Symp. Digit. Forensic Secur. (ISDFS)*, pp. 1-4, Mar. 2018.

15. A. Mahindru and A. L. Sangal, "MLDroid—Framework for Android malware detection using machine learning techniques", *Neural Comput. Appl.*, vol. 33, no. 10, pp. 5183-5240, May 2021.

16.X. Su, D. Zhang, W. Li and K. Zhao, "A deep learning approach to Android malware feature learning and detection", *Proc. IEEE Trustcom/BigDataSE/ISPA*, pp. 244-251, Aug. 2016.

17.K. A. Talha, D. I. Alper and C. Aydin, "APK auditor: Permission-based Android malware detection system", *Digit. Invest.*, vol. 13, pp. 1-14, Jun. 2015.

18.A. Mahindru and P. Singh, "Dynamic permissions based Android malware detection using machine learning techniques", *Proc. 10th Innov. Softw. Eng. Conf.*, pp. 202-210, Feb. 2017.

19.U. Pehlivan, N. Baltaci, C. Acarturk and N. Baykal, "The analysis of feature selection methods and classification algorithms in permission based Android malware detection", *Proc. IEEE Symp. Comput. Intell. Cyber Secur. (CICS)*, pp. 1-8, Dec. 2014.

20.M. Kedziora, P. Gawin, M. Szczepanik and I. Jozwiak, "Malware detection using machine learning algorithms and reverse engineering of Android Java code", *Int. J. Netw. Secur. Appl.*, vol. 11, no. 1, pp. 1-14, Jan. 2019.

21.X. Liu and J. Liu, "A two-layered permission-based Android malware detection scheme", *Proc. 2nd IEEE Int. Conf. Mobile Cloud Comput. Services Eng.*, pp. 142-148, Apr. 2014.

22.*Permission-Based Android Malware Detection | Semantic Scholar*, Oct. 2021, [online] Available: https://www.semanticscholar.org/paper/Permission-Based-Android-Malware-Detection-Aung Zaw/c8576b5df33813fe8938cbb19e35217ee21fc80b.

23.D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon and K. Rieck, "Drebin: Effective and explainable detection of Android malware in your pocket", *Netw. Distrib. Syst. Secur. Symp*, 2014.

24.H. Cai, N. Meng, B. G. Ryder and D. Yao, "DroidCat: Effective Android malware detection and categorization via app-level profiling", *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 6, pp. 1455-1470, Jun. 2019.

25. P. Rovelli and Ý. Vigfússon, "PMDS: Permission-based malware detection system" in Information Systems Security, Cham, Switzerland:Springer, vol. 8880, pp. 338-357, 2014.