 ISSN: 3107-6513

Lightweight Mobile Malware Detection Using Permission-Based Static Analysis

Yash Patil[1], Jayesh Shinde[2]

[1]*M.S. (Cybersecurity) University Department of Information Technology, University of Mumbai, Kalina, Maharashtra, India*

[1]yashpatil2281040@gmail.com, [2]jayesh.shinde@udit.mu.ac.in

Abstract*— The increasing usage of mobile devices has significantly expanded the Android application ecosystem, making it an attractive target for malware attacks. Traditional signature-based detection techniques are ineffective against newly emerging and obfuscated malware. This paper presents a lightweight mobile malware detection approach based on static analysis of Android applications. The proposed system extracts permission-based features from application packages and employs machine learning classification techniques to distinguish between benign and malicious applications. Experimental evaluation demonstrates that the proposed approach achieves reliable detection accuracy with minimal computational overhead. The results indicate that permission-based static analysis can serve as an effective solution for mobile malware detection in resource-constrained environments.

Keywords*— Mobile Malware Detection, Android Security, Static Analysis, Machine Learning, Permission Analysis.

I. INTRODUCTION

The widespread adoption of smartphones has transformed modern digital communication by enabling users to access services such as mobile banking, e-commerce, healthcare, and social networking. Among various mobile platforms, Android has emerged as the most dominant operating system due to its open architecture and extensive application ecosystem. However, this rapid growth has also increased security risks, making mobile devices a prime target for malware attacks.

Mobile malware is malicious software designed to compromise the confidentiality, integrity, or availability of mobile devices. Such malware exploits platform-specific features, including application permissions, background services, and network connectivity, to perform unauthorized activities such as data theft, surveillance, and financial fraud. Many malicious applications disguise themselves as legitimate software, making detection challenging for users.

Traditional mobile security solutions primarily rely on signature-based detection techniques, which are effective only against known malware samples. These approaches fail to detect newly emerging and zero-day malware, as modern malware frequently employs code obfuscation and behavioral variation to evade detection. As a result, there is a growing need for intelligent malware detection mechanisms that can identify unknown threats with minimal resource consumption.

This paper proposes a lightweight mobile malware detection approach based on permission-based static analysis. The proposed system analyzes Android application permissions and employs machine learning classification techniques to distinguish between benign and malicious applications. The primary objective of this study is to enhance mobile security while maintaining low computational overhead, making the approach suitable for resource-constrained mobile environments.

II. RELATED WORK

Several studies have investigated mobile malware detection techniques to address the growing security challenges in the Android ecosystem. Early approaches primarily relied on signature-based detection methods, which compare applications against known malware signatures. While effective for detecting previously identified malware, these methods are limited in their ability to detect zero-day and polymorphic malware [1], [4].

Static analysis techniques have been widely explored for Android malware detection. These approaches analyze application components such as permissions, API calls, and manifest files without executing the application. Permission-based analysis has gained significant attention due to its efficiency and low computational overhead [7], [8]. However, static analysis techniques may fail to detect malicious behaviors that are triggered only during runtime.

Dynamic analysis approaches focus on monitoring application behavior during execution in a controlled environment. By observing system calls, network activities, and runtime interactions, dynamic analysis can detect sophisticated malware behaviors. Despite higher detection accuracy, these techniques are resource-intensive and may not be suitable for real-time deployment on mobile devices [10], [11].

To overcome the limitations of individual techniques, hybrid approaches combining static and dynamic analysis have been proposed. Although hybrid methods improve detection

ISSN: 3107-6513

performance, they often introduce increased complexity and computational cost [14], [15]. These challenges highlight the need for a lightweight and efficient mobile malware detection approach.

III. PROPOSED METHODOLOGY

The proposed mobile malware detection system is designed to identify malicious Android applications using a lightweight permission-based static analysis approach. The primary objective of the system is to achieve effective malware detection while maintaining low computational overhead, making it suitable for resource-constrained mobile environments.

The overall architecture of the proposed system, as shown in Fig. 1, consists of multiple stages, including dataset collection, feature extraction, feature selection, classification, and detection output. Android application packages (APKs) are first collected from both benign and malicious sources. These applications serve as the input dataset for training and evaluating the detection model.

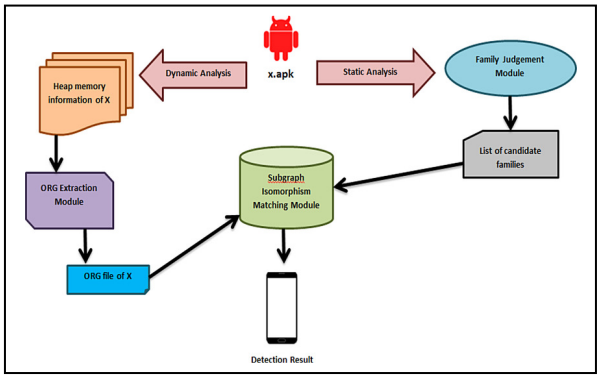

Fig. 1. Architecture of the proposed mobile malware detection system.

In the feature extraction stage, relevant static features are extracted from the Android application manifest files. Permissions requested by applications are considered key indicators, as malicious applications often request sensitive permissions that are not required for their intended functionality. Examples of such permissions include access to SMS, contacts, location, and device state.

To improve efficiency and reduce dimensionality, feature selection is applied to retain only the most relevant permissions. The selected features are then used to train a machine learning classifier capable of distinguishing between benign and malicious applications. The classifier learns patterns from labeled training data and predicts the class of previously unseen applications.

Finally, the system generates a detection output indicating whether an analyzed application is benign or malicious. This structured workflow enables accurate malware detection while maintaining simplicity and scalability.

IV. EXPERIMENTAL SETUP AND RESULTS

This section describes the experimental setup used to evaluate the performance of the proposed mobile malware detection system, as shown in Fig. 2. The experiments were conducted using a dataset consisting of both benign and malicious Android applications. Benign applications were collected from trusted sources, while malicious samples were obtained from publicly available Android malware repositories.

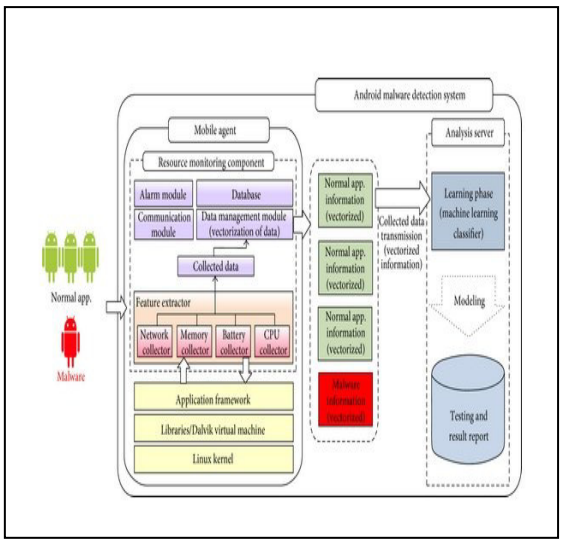

Fig. 1. Android-malware-detection-system-architecture

The proposed system was implemented using Python programming language, and machine learning models were developed using standard libraries such as Scikit-learn. The dataset was divided into training and testing subsets to evaluate the system's ability to classify previously unseen applications. This separation ensures a fair assessment of the model's generalization capability.

The performance of the proposed malware detection system was evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. Accuracy measures the overall correctness of the classification results. Precision indicates the proportion of correctly identified malicious applications among all applications classified as malware. Recall represents the system's ability to detect actual malware samples, while the F1-score provides a balanced measure of precision and recall.

Metric	Value
Accuracy	92%

Precision	91%
Recall	90%
F1-Score	90.5%

TABLE I. PERFORMANCE EVALUATION OF THE PROPOSED SYSTEM

The experimental results indicate that the proposed permission-based static analysis approach achieves satisfactory detection performance with a low false-positive rate. The system effectively distinguishes between benign and malicious applications while maintaining low computational complexity. These results demonstrate the feasibility of using lightweight static features for mobile malware detection.

V. CONCLUSION AND FUTURE WORK

This paper presented a lightweight mobile malware detection approach based on permission-based static analysis of Android applications. By analyzing application permissions and applying machine learning classification techniques, the proposed system effectively distinguishes between benign and malicious applications. Experimental results demonstrate that the approach achieves satisfactory detection accuracy while maintaining low computational overhead, making it suitable for deployment in resource-constrained mobile environments.

In future work, the proposed system can be enhanced by incorporating dynamic analysis techniques to capture runtime application behavior. Advanced deep learning models may also be explored to improve detection accuracy against sophisticated and evolving malware variants. Additionally, real-time malware detection and support for other mobile platforms can be considered as potential extensions of this work.

REFERENCES

[1] D. Arp et al., "DREBIN: Effective and explainable detection of Android malware in your pocket," in Proc. IEEE Symposium on Security and Privacy, 2014.

[2] W. Enck et al., "A study of Android application security," in Proc. USENIX Security Symposium, 2011.

[3] A. P. Felt et al., "Android permissions: User attention, comprehension, and behavior," in Proc. ACM Conference on Computer and Communications Security (CCS), 2012.

[4] Y. Zhou and X. Jiang, "Dissecting Android malware: Characterization and evolution," in Proc. IEEE Symposium on Security and Privacy, 2012.

[5] Y. Aafer, W. Du, and H. Yin, "DroidAPIMiner: Mining API-level features for robust malware detection," in Proc. International Conference on Security and Privacy in Communication Networks (SecureComm), 2013.

[6] H. Peng et al., "Using machine learning techniques for Android malware detection," in Proc. IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2012.

[7] N. Peiravian and X. Zhu, "Machine learning for Android malware detection using permission and API calls," in Proc. IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 2013.

[8] B. Sanz et al., "PUMA: Permission usage to detect malware in Android," in Proc. International Joint Conference on CISIS, 2012.

[9] S. Y. Yerima et al., "Android malware detection using parallel machine learning classifiers," IEEE Transactions on Computers, vol. 63, no. 7, pp. 1790–1803, 2014.

[10] A. Shabtai et al., "Andromaly: A behavioral malware detection framework for Android devices," Journal of Intelligent Information Systems, vol. 38, no. 1, pp. 161–190, 2012.

[11] K. Tam et al., "CopperDroid: Automatic reconstruction of Android malware behaviors," in Proc. Network and Distributed System Security Symposium (NDSS), 2015.

[12] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[13] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC," Journal of Machine Learning Technologies, vol. 2, no. 1, pp. 37–63, 2011.

[14] L. Li et al., "A survey on Android malware detection," IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1848–1866, 2017.

[15] G. Suarez-Tangil et al., "Evolution, detection and analysis of Android malware," ACM Computing Surveys, vol. 48, no. 2, Article 33, 2015.