# Android Malware and Analysis

**Rajiv Chourasiya**

**Himanshu Kumar Sengar**[1] **(22SCSE2130037 )**

**AKASH KUMAR**[2] **(22SCSE2130034)**

**Rinku Maurya**[3] **(22SCSE2130036)**

**[1] School of Computer Science and Engineering, Galgotias University, Greater Noida ,UP, India**

**[2] School of Computer Science and Engineering, Galgotias University, Greater Noida, UP, India**

**[3] School of Computer Science and Engineering, Galgotias University, Greater Noida,UP, India**

## ABSTRACT

Android malware analysis is a vital cybersecurity discipline involving the meticulous dissection of malicious software targeting Android devices. This process encompasses sample acquisition, static and dynamic analysis, code inspection, behavioral profiling, and reverse engineering. Analysts identify malware intent, vulnerabilities, and evasion techniques. YARA rules and signatures aid detection, while comprehensive reports guide mitigation. This rigorous analysis enhances our understanding of Android malware threats, fosters knowledge sharing, and bolsters defenses in the ever-evolving landscape of mobile cybersecurity.

## KEYWORDS

Cyber security, malicious software, techniques for detecting, detection rate, hybrid, dynamic, and static detection for Android mobile devices.

## 1. INTRODUCTION

With the most users around the globe, Android has overtaken other operating systems in the field of mobile telecommunications. This most popular mobile operating system supports a huge number of financial apps, including internet and mobile banking. Android phones also retain other private data, like passwords, usernames, and medical records. Android's technological development has drawn malware writers in proportionally, who are working harder every day to establish themselves financially by developing malware applications that can directly breach the security of mobile devices running Android, steal victims' personal information, or demand ransom in exchange for bitcoin once an effective attack has been carried out.

### 1.2 Malware

Despite the differences in these definitions, they ultimately converge on the idea of malicious programmes that have bad intentions.

According to [1-3]'s study, malware is merely a term for harmful software. By this definition, malware is any software component created with malicious intentions. Many of the attack and damage trajectories used by malware are not covered by this concept.

### 1.3 Attacks by Malware on Android Smartphones

The term "Android malware" refers to malicious software that is primarily intended to harm. Most Android users download free apps without carefully evaluating whether or not those apps are actually given by Google. Many people don't enable the application permission monitor [9, 47] on their Android devices, despite the fact that doing so might assist confirm whether or not installed apps have been verified and vetted by Google bouncer [10] before being installed. According to the research published in [11, 42–43], downloading applications from untrusted sites is a significant attack vector for malware infection on mobile devices. In addition, it's a smart security practise to be watchful while giving apps authorization rights.

## 2. RELATED WORK

It is now important to do research on mobile security. Mobile technologies are the subject of continuing study in a number of areas, including design, vulnerability, threats, and detection methods. Billions of dollars are being invested in this area by several security-related firms. Some of them served as the building blocks for this essay's analysis. The two main Android malwares that were examined in this research were DroidScope discovered DroidKungFu and DroidDream malware. . The study's findings showed that the method was effective at preventing resistance.

With 20% of the samples being unclassified, the study found a self-replicating Trojan with 90% accuracy. The study found that an investigation of Android mobile device memory dumps might yield valuable malware information. It is simple to reveal hidden codes that are ready to decode at the right time. However, the research was unable to offer any examination into looking for malware properties that are important for forensic and security analysis.

## 3. METHODOLOGY

In this research, an organised evaluation will be conducted using the detailed malware detection methodology described by [13]. The investigation of current research on Android malware detection methods is the goal of the analysis. The study questions, selection criteria for papers, data source are included in the methodological subheadings that follow. These were all included in Table 1.

### 3.1 **Source of Data**

Reliable academic research databases that cover computational fields and have a strong record of publications are good places to get trustworthy data sources. The major source of data for the study was journals and conference papers from such sources. Because they are deemed untrustworthy, papers from questionable sources like Wikipedia were not taken into consideration.

But because of its reputation, a reliable blog like SAN blog might be used to source data. The technological difficulties, constraints, and advantages of the malware detection methods utilised for Android are the major pieces of information that we plan to mine. Experimental findings and the methodology used by each study will not also be disregarded.

## 4.TECHNIQUES FOR DETECTING ANDROID MALWARE

The following malware detection methods will be looked at for this survey's purposes:

### 4.1 Dynamic Detection Techniques

It is possible to track interactions between Android-based applications and the device OS thanks to system calls. It examines how malware interacts with mobile services and resources, including behaviours related to the network, the operating system, packages, and location. It is advised that the code execution be done in a cybernetic environment to protect the safety of the experimental apparatus (physical device).
the study of [19] used this detection approach. Malware on such apps was accurately discovered with 96% precision by the random forest classification method when using the ServiceMonitor technique. Malware was shown to have 67% accuracy in its information retrieval, including phone IMEI.

### 4.2 Static Detection Techniques

Static malware detection does not run any dangerous code; instead, it only uses the characteristics of the malware abstraction layer. The APK format is used for Android apps. A zip packet often contains this. There are resources for all Android files, directories, and applications. In order to mine features from the apk files fomeaning identification, reverse engineering is frequently used. The "AndroidManifest.xml" manifest file should be taken into account initially while looking for suitable feature extraction. Permission vector features for installation, locations, battery optimisation, and other features are included in this manifest file.

### 4.3 Methodologies for Hybrid Detection

This method combines the advantages of static and dynamic approaches to analyse malware and produce a more reliable detection result. Essentially, the training and detection stages of a hybrid detection method to malware detection may be carried out using static and dynamic approaches.Given that the benefits of both approaches are integrated, this seems to offer a higher detection rate than dynamic and static strategies. A DroidDetector [51] model was created and trained with a few algorithms for android malware detection using the deep learning component of artificial intelligence [21]. 192 examples of both benign and malicious Android code were gathered for training using the hybrid method. With a 0.0021% difference across the techniques utilised, the model produced detection results with a 96.60% accuracy.

Some Android malware is primarily meant to gather data about system calls, filesystems, location, and camera photos. The user of the device is exposed to physical harm and information security risks when using a malicious programme with this target. His system files may be abused for money or other benefit, and it would be simple to find him and harm him. This was shown in the research of [25] using a modest dataset of Android malware.

## 4.4 Permission-Based Methods for Detection

This method entails a thorough study of every packet of network data that originates from the http server.The kind of data that an application or device is sending or receiving from a distant server can be ascertained by analyzing these packets. Certain mobile spyware, particularly when utilizing an unsecured communication channel,limit the disclosure of personally identifiable information (PII) to a malicious URL, location, mac address , images , IMEI , and IMSI. Malware of this kind does not damage the host device in a noticeable or noticeable way. Information leaks may be discovered when this communication is recorded and studied by tools like Wireshark.

It has been shown that such malware's intercepted data may be found when the right strategy is used. Using this method to examine malware, one should concentrate on features such as communication protocol and apk files [29].

## 4.5 Emulation Based Detection

In order to keep malware samples apart from the device's actual physical resources, this strategy calls for the provision of a simulated ecosystem via an emulator. The Android OS or hardware can be used for simulation. When malware is executed within a mobile operating system, however, identification becomes considerably more difficult.

By collecting the dex file and turning it into a human-understandable format, malicious apps in the sandbox system may be identified. With this method, malware that elevates privileges [35] and zero-day malware [34] may be successfully caught. The virtual aspect of the environment, however, makes certain viruses aware and makes it more likely to avoid detection.

## 5. DISCUSSION

Basic conclusions about the detection strategies were drawn from this study's comparative examination. One of the fundamental constraints found in the explored strategies was the use of a tiny malware dataset. Due to the sample size's inability to encompass all malware family boundaries, a thorough evaluation of the detection effectiveness is hampered. While the approach may have appeared to work well with this sample size, the reverse is really true when applied to a bigger dataset. With minimal optimisation, this might lead to an unbalanced ratio.

It is clear that some viruses would likely be able to evade detection by comparing different system information with strings like "VMware" or "QEMU," which would allow them to determine whether a sandbox environment was present. Sys_vendor file-testing malware families or samples will be able to identify the sandbox analysis environment, particularly if the analysis is carried out with root privileges.

## 6. EVALUATION

The study of Android malware detection methods is important for creating a useful detection tool that incorporates both the advantages and disadvantages of all the investigated methods. We used the static detection methods to harvest Android metadata and other artifacts from malicious Android applications. Malicious interaction with the dex files at the Dalvik layer offers a dynamic approach to detection. A hybrid approach that increases detection accuracy is created by combining static and dynamic methods.Additional techniques under investigation include detection based on emulation and content. It is important to highlight in this study that a variety of trained algorithms may be used to address both dynamic and static procedures in various ways (see Appendix). A dynamic method can get around detection problems like malware fitting and oligomorphism.

the requirement for hand-engineering is eliminated by using an opcode sequence technique. The issue of malware encryption could not be solved by this, though. When this method is applied, malware can be obfuscated to avoid detection. Using CFG may result in erroneous results if certain human interference factors are present. Even though it is very scalable, It is possible for malware to load and spread on CFG. Using a content-based strategy, large malware datasets can be quickly run. Android mobile forensic faces a variety of difficulties, including malware detection, as noted by [36, 37].

## 7.CONCLUSION

A comparison of several Android mobile malware detection techniques was presented by this study. Through a process of critical review, the research was able to pinpoint the weaknesses and advantages of each of its detection approaches. The investigation's findings corroborate the assertion that methods designed to identify Android malware don't always produce 100% accurate detection results.

This section of the study offers a critical assessment of the articles under consideration. The goal of this comparison analysis is to provide a clear knowledge of the advantages and disadvantages of the chosen detection techniques that were discovered via research. A comparison of detection methods with a primary focus on detecting Android malware detection methods and comparing their respective detection methodologies, detection precision.

## 8. FUTURE WORK

Given that some of the research's potential gaps have been highlighted, there are numerous chances and opportunities to further the work reported in this study.

First off, rather than only implementing machine learning algorithms, Better detection simulation using AI techniques and deep learning technologies might fortify hybrid and dynamic detection frameworks. Improvements to hybrid detection techniques can assist boost code coverage and sample streamlining effectiveness, as mentioned in section 6 (see Table 1).

Furthermore, the integration of hybrid detection emulators with genuine Android phones would help address the problem of complex malware, such polymorphic malware, being able to recognize virtual environments, and avoiding detection of VM-ware. When this gap is filled in the future, the accuracy of this solution will increase.

## REFERENCES

[1] Li., F., Tegawendé, Y., Klein, D., Traon, L.: Understanding android app piggybacking: A systematic study of malicious code grafting. In: Tran. 2017. IEEE conference on Information Forensics and Security, vol. 12, pp. 1269--1284 (2017).

[2] Abdul, R., Daud, M., Mohamad, M.: Securing sensor to cloud ecosystem using internet of things (iot) security framework. In: Proc. 2016. ACM International Conference on Internet of things and Cloud Computing, pp. 79, ACM press (2016).

[3] Yan, C., Zahedi, F.: Individuals' Internet Security Perceptions and Behaviors: Polycontextual Contrasts Between the United States and China. Mis Quarterly, vol. 40, no. 1, pp. 205--222 (2016).

[4] Zhaoguo, W., Li, C., Yuan, Y., Xue, Y.: DroidChain: A novel Android malware detection method based on 9behaviour chains: Pervasive and Mobile Computing, vol.32, pp. 3--14 (2016).

[5] Songyang, W., Wang, P., Zhang, Y.: Effective detection of android malware based on the usage of data flow APIs and machine learning: Information and Software Technology, vol. 75, pp. 17--25 (2016).

[6] Andrea, S., Sgandurra, D., Dini, G., Martinelli, F.: Effective and efficient behaviour-based android malware detection and prevention: IEEE Transactions on Dependable and Secure Computing, (2016).

[7] Anurag, S., Kumar, D., Chanana, L.: An end to end security framework for service-oriented architecture: In Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), IEEE International Conference, pp. 475--480 (2017).

[8] Kaspersky, https://www.kaspersky.co.uk/resource-center/preemptive-safety/what-is-malware-and-how-to-protect-against-it.

[9] Muhammad, I., Vallina-Rodriguez, N., Seneviratne, M., Paxson, V.: An analysis of the privacy and security risks of android vpn permission-enabled apps. In: Proc. 2016. ACM conference on Internet Measurement, pp. 349--364 (2016).

[10] Wenrui, D., Liu, X., Li, Z., Zhang, K.: Evading android runtime analysis through detecting programmed interactions. In: Proc. 2016. ACM Conference on Security & Privacy in Wireless and Mobile Networks, pp. 159--164 (2016).

[11] Yasuyuki, T., Goto, A.: Analysis of malware download sites by focusing on time series variation of malware. Journal of computational science, vol. 22, pp. 301--313 (2017).

[12] Android malware infection, https://www.lincolnshirelive.co.uk/news/local-news/check-your-apps-now-145-1876747.

[13] Hao, X.: Advanced Android Malware Detection Framework. MIT Press, Cambridge, MA (2013).

[14] Lok-Kwong, Y., Yin, H.: DroidScope: Seamlessly Reconstructing the OS and Dalvik Semantic Views for Dynamic Android Malware Analysis. In: Symp. 2017. USENIX security symposium, pp. 569--584 (2017).

[15] Anastasia, S., Gamayunov, D.: Review of the mobile malware detection approaches: Parallel, Distributed and Network-Based Processing (PDP). In: Proc. 2015. IEEE 23rd Euro micro International Conference, pp. 600--603(2015).

[16] Anusha, D., Troia, F. D., Visaggio, C. A., Austin, T. H., Stamp, M.: A comparison of static, dynamic, and hybrid analysis for malware detection. Journal of Computer Virology and Hacking Techniques, vol. 13, no. 1, pp. 1--12 (2017).

[17] Shuaifu, D. Y., Liu, T., Wang, T., Zou, W.: Behavior-based malware detection on mobile phone," In Wireless Communications Networking and Mobile Computing. IEEE International Conference, pp. 1--4 (2016).

[18] Latika, S., Hofmann, M.: Dynamic behaviour analysis of android applications for malware detection. In IEEE International Conference on Intelligent Communication and Computational Techniques (ICCT), pp. 1--7 (2017).

[19] Majid, S., Amini, M.: Android Malware Detection using Markov Chain Model of Application Behaviors in Requesting System Services" arXiv preprint arXiv:1711.05731 (2017).

[20] Ankita, K., Troia, F. D., Stamp, M.: Static and Dynamic Analysis of Android Malware: In ICISSP, pp. 653--662 (2017).

[21] Zhenlong, Y., Lu, Y., Xue Y.: Droiddetector: android malware characterization and detection using deep learning: Tsinghua Science and Technology, pp. 114-123. IEEE Press, (2016).

[22] Stamp, M.’ Anusha, D. F.: A comparison of static, dynamic, and hybrid analysis for malware detection,” Journal of Computer Virology and Hacking Techniques, vo. 13, no. 1, pp. 1--12 (2017).

[23] Lingwei, C., Hou, S., Ye, Y., Chen, L.: An Adversarial Machine Learning Model Against Android Malware Evasion Attacks: In Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data, pp. 43--55, Springer, Cham (2017).

[24] Shifu, H., Saas, A., Ye, Y., Chen, L.: Droiddelver: An android malware detection system using deep belief network based on api call blocks: In International Conference on Web-Age Information Management, pp. 54--66, Springer, Cham (2016).