Review: Android Privacy Leakage Detection using Static Analysis

Umar badamasi ibrahim, SaurabhSrivastwa, Manish Verma
School of Engineering & Technology
Sharda University, Greater Noida, Uttar Pradesh, India

Abstract:- Android is an open-source Operating System (OS) that is free and allows for a comprehensive understanding of its architecture. As a result, lots of manufacturers, including Samsung, Google Pixel, Sony, and Motorola, are using this system to create mobile devices (Mobile phones, smartwatches, and smart glasses). The use of an operating system leads to an expeditious growth in the number of Android users. On the other hand, unethical authors tend to create privacy devices for the sake of wealth or fame. Even though practitioners perform encroachment perception studies such as static analysis, this research examines some recently published articles from the year 2018 to early 2022. Based on the survey clearly stating that the android privacy leak uses the android characteristic, AmpDroid and FlowDroid also play a good role in detecting malware, later on, machine learning(ML) and deep learning algorithm is introduced to the system to find high accuracy and also listed the component and the stages of the analysis way to detect Android malware based on its high algorithm effect and accurate result.

Keywords:- *Android privacy detection, Static analysis, Machine learning, deep learning algorithms, stages, compone.*

I. INTRODUCTION

Android has grown to become the most popular mobile (OS) in the last decade, due to the spectacular rise of the mobile industry. So far, (OS) such as smartphones have accounted for more than 80% of the total market. [1] In the mobile (OS), Inter-Component Communication (ICC) is an android passing system that allows an application to communicate with each other. ICC's misemploy can lead to some major security issues, as well as data theft and privilege escalation attacks [2]. Previous researchers have proposed three detection ways to identify privacy, including static, dynamic, and hybrid techniques, to address the aforementioned concerns. Static analysis is a technique that is performed before the Android application is installed and is based on theAndroid Application Package (APK) source code. As a result, privacy can't change its behaviour during static analysis. On the other hand, dynamic analysis tests and assesses Android software in real-time. Finally, the final technique will search for all conceivable code and runtime defects using a combination of static and dynamic characteristics. The process feature selection is critical for detecting privacy since raw information may result in an erroneous result [3], when classifying applications, taking into account too many characteristics might incur computational overhead and take a long time [4]. Though Android has only been proposed for a few years, vast numerous researchers have focused on various difficulties relating to Android applications. Their objectives, such as discovering potential issues in apps, boosting application performance, and securing smartphone personal information from fraudulent apps, have led them to suggest a variety of ideas and techniques and develop a variety of models.Ordinary people seem to find flashing or rooting devices to execute the customized OS a difficult undertaking[5] and Additionally, several other issues such as the fragmentation issue must be addressed[6].

In this article, we explore the previous research/survey that has been done in static analysis that are already published from the well-known conference/journal from the year 2018 t0 2022 and based on the best-chosen papers we read and summarized their work and the proposed method used and the achievement and lack of later conclude, so basically we review the previous researches on the analysis not providing another method of analysis rather bring what to look into in the future of static analysis.

II. BACKGROUND

There's a need to comprehend several methodologies to select the optimum security architecture for analysing malicious programs and detecting information leakage. The following is some background information on information security and analytic methodologies in many sectors.

A. Information Leakage and its impact on the security

Depending on the circumstances, Information can be classified as sensitive or non-sensitive. The most recent smartphones are capable of transporting critical information about mobile phone users. The location of the Global Positioning System(GPS) module may be used to show the device's location on the map.DeviceID may be used to unambiguously pin out a mobile phone user. Wireless security of access points may be determined using network information from a wireless network. The majority of new Android phones come with various sensors built-in. The sensors contain data/information that is unique to the user. Additionally, to data acquired by physical sensors, data saved in smartphone storage, such ascontacts details, phone messages, subscriber identity module (SIM) card details, and camera photographs, is collected. can potentially give a lot of information about the individual If the user's sensitive information is disclosed, the user is subject to extra dangers connected with the criticality of data, such as GPS coordinates being exploited by an attacker to monitor the user's position. Terrorist groups spied on army locations based on information supplied by the spyware, according to one of the articles [9]. The attacker can utilize Wi-Fi-related information to determine the wireless security of the

Android user's operating Environment. A rogue domain or logs might be used by an Attacker to disclose sensitive information from Android smartphone users.

B. Static Analysis approach

It is a method for analyzing malicious activity in an application without running it. [7], The static analysis does not need any type the code to detect any malicious threat instead of this static analysis only examines the file or signs of the malicious procedure to find any kind of threat that had happened It is important because it identifies the libraries, packets, header files and the infrastructure. This investigation uncovers the flaws in code validation as well as concerns with cryptography implementation.[8],In static analysis the technical indicators are defined by such as file name, hashes, string, likes domains, IP address, and file header data used for the identification of any malicious intent. Some tools like disassemblers and network analyzers are used to detect the spyware/malware to collect the data without running it sincethe order that how malware work. The static analysis does not work on the coding process, which leads to some malware that isn't discovered. If the filecontains a string and after that, it loads the malicious things then the static analysis cannot detect it.

In the next section, we will go through the Dynamic analysis approach and its benefits and drawbacks.

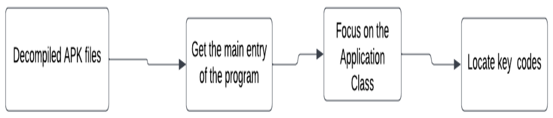
Decompiled APK files
Get the main entry of the program
Focus on the Application Class
Locate key codes

Fig. 1: Android apk

C. Dynamic analysis approach

The dynamicanalysis approach is based oncode, which is known as a sandbox to find out the malicious code to make the entire system safe and secure. Dynamic malware analysis works closed system to know all threats it always watches the System and doesn't let the attackers enter easily and infect the whole working enterprise. Dynamic malware analysis has had stages such as threat hunters and incident responders which have the elegant power to find the malicious act which helps them to reach the correct segment at minimal time and uncover the nature of the threat. On the other hand automatic process of the sandboxing whips to minimize the period andreverse the file to the developer to discover the malicious code. The main consequences of dynamic malware analysisarethat adversaries are smartand there is no boxtherewhich is a good analysis to determine the threat, the sandboxadversaries hide code into them which may remain dormant until conditions are met.The most common method for dynamic analysis are Fuzz testing, concolic testing, and search-based testing are the popular techniques.[10]

Dynamic methods analyse the behaviour of software by executing methods. By executing the software on a virtual computer and then examining the data created with packet capture tools, the software may be dynamically examined. The key advantage of employing dynamic analysis over software is the ability to follow the app's runtime behaviour, which is not available with static analysis. The graphic depicts one privacy sample's network activity as recorded by Wireshark. The leaked data is in plain text, so we can examine what kind of data the sample is leaking.To automate a validation process such as evolutionary algorithm and optimization computation, search-based testing uses a morpho optimized search approach[11].

D. Hybrid analysis

Hybrid analysis works on a combination ofthe characteristics of both static analysis and dynamic, static analysis cannot detect all type of malware sophisticated malware can easily passes through the static analysis and also some time the dynamic malware analysis code cannot detect some malicious code for that reason the hybrid malware analysis had made by combining the characteristics of the other analysis which is static and dynamic analysis techniques, the hybrid analysis gives the all set of the team the ability to detect the threat by using both of the analysis methods. Primarily because it can handle finding the maliciouscode in the hidden malicious code tract. Hybrid analysis to detect the most sophisticated malware and can cure it. One of the hand hybrid analysis use static analysis to monitor the pattern and the behaviour of the data and on the second hand if the malicious code run and find some changes in memory thentrails are transferred to the dynamic analysis and it will handle that to find the threat and after that analysts alter that to static analysis that some dump in memory than static analysis act on that and tries to find the threat.

E. Why Static Analysis Over Dynamic Analysis

Capturing encrypted network traffic generated by the samples makes it difficult to detect information leaking. Privacy can encrypt data before leaking it to a hostile domain, making it difficult for dynamic analyzers to identify information leakage. On the other hand, a static analyser can employ taint tracking to detect information flow from source to sink. Static analysis examines the contents of a single file as it resides on a disc rather than when it is detonated. It parses data, extracts patterns, properties, and artefacts, and flags abnormalities. Static analysis is impervious to the problems that dynamic analysis poses. It is far more efficient and cost-effective, taking only a fraction of a second. Static analysis may also be applied to any file because there are no unique criteria, environments that must be customized, or outgoing communications required from the file for analysis to take place.

F.COMPARING MALWARE ANALYSIS

Approach	Advantages	Disadvantages
Static analysis	It's faster and more secure. It finds weaknesses in the code at the exact location. It consumes low resources. High accuracy. It is multipath malware analysis.	It cannot detect unknown malware. It is time-consuming if conducted manually. It can't analyze encrypted malware. Automated tools do not support all programming languages
Dynamic analysis	Can analyze encrypted malware. It identifies vulnerabilities in run time environment. Can detect both known and unknown malware. It can be conducted against any application	Slow and unsecure. Automated tools produce a false positive and false negative. High resources consumption. It consumes lots of time.
Hybrid analysis	It has higher accuracy More effective than dynamic and static analysis	It consumes a lot of time compared to other analysis Complexity is high

Table 1: Comparing malware analysis

III. STAGES

A. Static Properties Analysis:

In static properties which have strings embedded in malware code, header details, metadata, embedded resource, hashes and so on. To create IOCs all this data are needed and they work in the y quick process because there is no need to run any code for analyses. Insight is gathered at the static properties analysisof all the segments if their investigation is needed and using the more comprehensive techniques is necessary and which step should be taken next.

B. Interactive Behavior Analysis:

Behaviour analysis works on the process of observing and interacting with a sample of malware that runs in a lab. It can be understood by the samples registry, process, network and file system which is also known as memory forensic which learns how the malware uses the memory. If there is any suspicious sample found that Cheney set the simulation to check their theory. In behavioural analysis, there is a requirement forthe best analyst with high skills because it is a time-consuming process and lots of patience can be required.

C. Fully Automated Analysis:

In fully automated analysis can access the malicious file very easily and simply. Analysis can be determined on the potential of the repercussions and it infiltrates the network and gains easy to read the report of the malware it provides a fast answer for security teams. For the large scale analysis of malware then fully automated analysis is the best way.

D. Manual Code Reversing:

Manual code reversing analysts it used the system of reverse engineer code by using debuggers tools, compilers, disassemblers, andsome of the tools to decrypt the encrypted data to determine the mind and logic behind the algorithms know any hidden suspicious thing has exhibited. It is a very rare skill and needs much time to execute. For sometimes if it is not done then the valuable insights can be missed and not found the malicious event.

IV. COMPONENTS

Malware has some of the components thatwork on it to perform the auspicious event.

It has seven processes which are payload, packer, persistence, propagation, communication, armouring, and stealth.

A. Payload:

The payload is the middle or core of the data where the malware infects the host's objectives. Most malware developers are most sincere about it and try to make it very well developed and of the best quality to operate because the attackers most try to infect it if the quality is worse development.

B. Packer:

Packerencapsulates payload it is like eggshell of the payload. It is needed to encapsulate becauseit was needed to hide from the malware detection application. Packers compress the payload to make it obfuscate to the payload to hide from the detection of the static analysis and signature base analysis. It will open the payload with the algorithm which is defined by the developers, many algorithms in the wild are used to develop or pack the payload.

C. Persistence:

It is the factor where the attackers use to persist to the host by infecting them and they try to take command even after the host logs off the system or reboots the system, the attackers use any way to persist the data by tampering it by registry autorun and the many more.

D. Communication:

In communication, malware infects the host on the basics to connect from the backside of the malware connector (c2,c2c,cnCNCthe connection is done for the many things such as the taking out the details, sniffing of data or trying to take access, and sometimesretrieves data to the developers to make a new command to do any action. They use most of the time to encrypt channels to hide their communication from the networking detection.

E. Propagation:

In propagation after getting control over a host it t, tries to infect other another machine to establishits foothold in the network. It always tries to find UA vulnerability in the other system and network to make exploit to make access the other host.

F. Armoring:

In armouring creators create their code so that it cannot be easily detected by the anti-malware analysis such as antivirus, sandbox, EDR, and many more. When the suspicious malware enters the system and starts itswork to make the stem infected and gets to know that there is any antivirus present in the environment it tries to benign its activities to avoid fire detection.

G. Stealth:

In the stealth process, the sun, spacious event or malware need tobe kept hidden from the detection of the malware and the antivirus, the stealth process can be done by the alter the file properties which makesit very complex like infecting the networking system and the new file and all these processes done by the rootkits, code injection, and so on.

V. TYPES

Nowadays days the use of introverts has grown very far and for that reason, the attackers also get the chance to steal the data from the users and this will increaseevery outcome from this situation there are some tools which are used to take an analysis on such malicious event, so here some of the malware analysis tools are mentioned.

A. Cuckoo sandbox automated:

Cuckoo sandbox automates malware analysis which provides essential feedbackon the file which are working in the remote environment it is always the path where the troves of the malicious data. Very easy to write and customized and processing stages which makes the very reasonable to use it. Some of itsfeatures it are that can analyze suspicious data, can identify easy targets, helps in cyber –warfare and so on.

B. Zeek network monitor:

It is a security analysis tool which free and which is developed in 1994 by Vern Paxson. It works on IDS (intrusion detection system) which uses to monitor the data or the network traffic to determine the threat occurring procedure. Was It developed to anale theregistered network, live network traffic, and trace the files which may vulnerability. It can take seat sections attending the events are starting to start the alert, sending an email the l, giving the command, and callingzeek script. There are some important features of these tools are that they use the best data sources, uses the light sensor from the core value.

C. Netcat Dynamic Analysis tool:

This tool is used for the network connecting from the TCP and UDP to write the overall connection and to apply by using this tool. It is used for the scanning of the port, tunnelling, proxying and so on and it is also known as the swiss army knife. It works on the dynamic analysis which can be done on any of the networks where the analysis method is required, it is also used for inbound and outbound on any port and anyone can use it for listening also by joining the client-server Some of the features of these tools are that they use mostly for port scanning, port forwarding and many more.

D. Resource Hacker Malware Analysis Tool:

This tool is used for observing and extracting the windows file it is basically working the 32 and 64 bits technology, it uses to scan any icon and the bit maps of the icon and for any future use purpose, you can save it. The hacker may inject the malware data into images, videos, and text and sometimes it is covered in the DLL and OCX file anywhere. It cannot be saved and can give inappropriate results but it gives you stability and it is best to try. Some of the features of these tools are that they can modify the system, u of strings, resources can be exported, icon resources, and many more.

E. Dependency Walker Analysis:

These tools use as free services and can use to analyse or scanning of the files of 32 bit and 64 bit of the windows system and it is used for the making of hierarchical tree modules of all modules. It gives the minimum set of files with all other information and the path of the files and also provides all other needy information of the file, it gives the version numbers, debug data, address, and many more. It also helps in commonly mistakes such as importing/exportingmistakes, mismatches, module crashing and many more. It also helps in troubleshooting errors in the executing modules. Some of the important features of this tool are that it can easily detect the missing file, detect the curricular dependency error, and also helps to detect the missing file.

Roy et al. [7] have written an article on app vetting which is a system of Flow Droid and Amandroid that is used to complete the complete theory of static analysisand it's stated that the number of behaving datafailing and the number of false signals is the categories of faults that a static analysis tool tries to reduce. When generating the supervision flow graph, a static code analysis tool aims to prevent over-approximation (i.e., erroneous vertices just on the graph that generate anomalies or lower precision) and also under-estimation (i.e., erroneous vertices on the plot that generate near misses or less accuracy) (which leads to missed behaviours or lower recall). There is an exchange between precision and recall when using a static analytical model. It is critical for the lots and enhancement of research in the field to conduct a fair comparative review of existing tools. It's difficult to choose (or develop) an unbiased app benchmark that doesn't give any tool an undue edge. On several programmes, such as DroidBench and ICC-Bench, the precision and recall of FlowDroid and Amandroid are investigated. As a result, a quantitative comparison of these tools can't be done based on those indicators in this post. Instead, it can't put the tools to the test on a set ofcarefully crafted apps to see if they can detect various sorts of data leaks.

In the year 2019, Shrivastava& Kumar [8] performed research for Intent and permission modelling for android malware detection, the experiments with the combination of Android intents and permissions have shown a high rate of privacy detection, and The outcomes of the sensitivity analysis were excellent and effective. Also, it showed the findings by combining Android intents and permissions to show that the attributes do not coincide.and the results have shown Android Intent is a more reasonable trait in detecting malware. Based on the best results, it's clear that Android intent is more efficientin privacy detection than Android permission. Also, the risk was projected in terms of high, medium, and low, with the positive number representing clean and the negative value representing privacy. The proposed model can be used to analyse an unknown application.

Pan et al. [9] have justified and reviewed a paper that collects the data from previous studies from 2014 to 2020, it stated that from the data collected the division android privacy/malware detection is divided into four groups based on the application's features which are Android characteristic-based technique, opcode-based technique, programme graph-based technique, and symbolic execution-based method. Then using experimental proof. Evaluating static analysis's privacy detection capability and comparing the Android efficiency of various models of privacy detection finally it's shown to be effective in detecting Android privacy.

Furthermore, the results show that when it comes to identifying Android security, the neural network beats the non-neural network model. Static analysis, on the other hand, still faces numerous challenges and it is necessary to bring out some best ideas. Based on current research, techniques for improving Android privacy detection have been developed. According to the Single Lens Reflex (SLR), the most The android trait is a regularly utilisedstatic analysis approach in Android identity discovery; in empirical investigations, datasets libraries like Drebin and Genomics comprise the highest part. Apktool is commonly used as a static code assistance tool in most investigations. The most common feature reduction technique is Information Gain (IG) and the most useful features are permissions and sensitive API calls. The (ML) model accounts for the lion's share of all models used. And accuracy is the most commonly used performance measure; empirical evidence shows that static analysis techniques are the best method and system for malware, Jeon et al. [10] have researched android privacy leakage where he proposed a system of Repackaging with Malicious Code Injected (ReMaCi) a threat model that uses the repackaging assault, dubbed the ReMaCi attack, to try to expose users' information identified that 50 per cent of the top By analysing the top 8,546 applications installed from the Play Store, we discovered that they are vulnerable to the ReMaCi attack. As a result, AmpDroid was present, a revolutionary, For minimizing crucial data thefts, an automatic static pro technique is used.

AmpDroid detects impervious data flow and isolates the sensitive data-handling code from an application. AmpDroid's privacy and efficiency are assessed, and its utility is determined by matching to other obfuscated essential tools. AmpDroid can only prevent leaks of data by generating Multi-Layer Bytecode, using the real-world data AmDroid wasn't able to identify 50 per cent of the application, to protect the data some scheme used is class encryption and strong encryption. AmpDroid failed to analyse around half of the data in the experiment and only started running about a third of it. Even though their codes are heavily obfuscated, sensitive data flows must identify on unanalyzable applications to secure sensitive data. However, it recognises that statically evaluating obfuscated apps is a difficult task, and will address this issue in the future.

Paul et al. [11] Have worked extensively in the realm of detecting privacy leaks where a permission-induced risk interface MalApp is proposed to highlight infringement of privacyarising from giving authorization when installing the app to protect users' privacy. It is made up of several steps that do the static method relies on the classification of the app. To generate a Boolean-valued permission matrix, the retrieve of the app relies on a reverse engineering system, permissions are ranked to determine the most dangerous permissions in each category. Third, The effectiveness of the provided technique was tested using a data set of 404 innocuous and 409 malicious applications. using machine learning and ensembling approaches. According to empirical testing, the proposed method has a malware detection accuracy of 98.33% in the best-case scenario and 98.33% in the worst-case scenario. The interface's feature is that any software can be used. The interface's highlight is that static analysis may also classify any software as benign or malicious before it started. a method of reverse engineering in the form of static analysis was used to classify apps by subcategory. Permission ranking utilising the correlation coefficient was used to determine the dangerous permissions across the category. Numerous machine learning algorithms and granulating algorithms were utilised to assess the feasibility of the recommended technique, and the bounding limits put forward affiliate capability to find malicious software that isn't well-known. Random Forest categorised the Books general category at a rate of 73.33 per cent on the sparse dataset. Furthermore, the data demonstrate a total precision rate of 88.2 per cent.

VI. LITERATURE REVIEW

Reference Key	Key Contribution	Set Back
[12]	Uses a vetting approach to identify the strength and weaknesses of the system by analyzing the whole code	It is so limited in finding the weakness
[13]	proposed intent and permission modelling system to get a high privacy detection analysis and sensitivity analysis	It failed to use the system of (ML) and deep learning algorithms so the accuracy was low.
[14]	After collecting 98 studies, it shows that the neural network model performs better than the non-neural network model by analysing and comparing primary studies.	Need more improvement in the field
[15]	It proposed (ReMaCi) threat of AmpDroid, it evaluated 8546 automated static anti-analysis tools for preventing critical information breaches	AmpDroid failed to analyse around 50% of the dataset and only started running about a third of it.
[16]	The proposed mechanism collected data set of 404 benign and 409 malicious apps to test the efficacy (ML) and ensembling approaches. And reaches the overall accuracy rate of 88.2 per cent.	The heuristic value was so less which unable to find the highest accuracy in the system

Table 2: A survey of existing work

VII. CONCLUSION

Static analysis is an effective way of detecting android privacy leaks. Where you can able solve the problem before executing the program.

In this paper, we briefly survey the existing work on android privacy leakage detection using static analysis and find out that many approaches are indicated for detecting personal data/information, we choose our survey based on recent years of studies from (2018 to 2022), which we analyse based on static analysis and the preliminary result shows that the effective way to solve the problem of android privacy is by enlarging the scales of static analysis and identifying the threat at the backend, where we find out that most researchers use (1)AmpDroid and FlowDroid to detect privacy leakage, (2)app vetting technique (3) android characteristics (3) permission and intent protocol and other android apk tools, it stated that static analysis is still the best way of identifying and protecting privacy in android by the use of its features based on the data collected.

VIII. FUTURE SCOPE

The review of the previous thesis revealed that there are numerous future opportunities for static analysis to detect privacy attacks in Android. that Android privacy leak detection on static analysis still faces some challenges due to incapability to solve the entire problem which in the future we'll focus more on Machine Learning (ML) and deep learning algorithms for the adequate and fastest result. However, the static analysis method is incapable of solving issues such as code obfuscation and a high rate of false alarms analysis approach to solve the problem in the future.

ACKNOWLEDGEMENT

First and foremost we would like to thank Mr Manish Verma who took time out of his busy academic crisis to support and guide us through the completion of this project as well as provide necessary information regarding the project. My thanks to my great friends and colleagues who support and motivate me through the process and a big thanks to the Sharda University, CSE department for monitoring our progress and arranging all facilities to make life easier.

REFERENCES

[1.] Zhou, W., Zhou, Y., Jiang, X., &Ning, P. (2012). Detecting repackaged smartphone applications in third-party Android marketplaces. Proceedings of the Second ACM Conference on Data and Application Security and Privacy - CODASKY '12. https://doi.org/10.1145/2133601.2133640
[2.] Hoang, N. H. (2016). Poster. Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services Companion - MobiSys '16 Companion. https://doi.org/10.1145/2938559.2948874
[3.] Reddy, M. K. (2019). Malware detection & prevention in Android mobile by using significant permission identification & machine learning. International Journal for Research in Applied Science and Engineering Technology, 7(12), 142–149. https://doi.org/10.22214/ijraset.2019.12024
[4.] Li, G., &Geng, J. (2019). Characteristics of raw multi-GNSS measurement error from Google Android Smart Devices. GPS Solutions, 23(3). https://doi.org/10.1007/s10291-019-0885-4
[5.] You, W., Liang, B., Shi, W., Zhu, S., Wang, P., Xie, S., & Zhang, X. (2016). Reference hijacking. *Proceedings of the 38th International Conference on Software Engineering*. https://doi.org/10.1145/2884781.2884863
[6.] Felt, A. P., Chin, E., Hanna, S., Song, D., & Wagner, D. (2011). Android permissions demystified. *Proceedings of the 18th ACM Conference on Computer and Communications Security - CCS '11*. https://doi.org/10.1145/2046707.2046779
[7.] Shahriar, H., Islam, M., &Clincy, V. (2017). Android malware detection using permission analysis. SoutheastCon 2017. https://doi.org/10.1109/secon.2017.7925347
[8.] Toffalini, F., Sun, J., & Ochoa, M. (2018). Practical static analysis of context leaks in Android

Applications. *Software: Practice and Experience*, *49*(2), 233–251. https://doi.org/10.1002/spe.2659

[9.] Swetha, K., &V.D.Kiran, K. (2018). Survey on mobile malware analysis and detection. International Journal of Engineering & Technology, 7(2.32), 279. https://doi.org/10.14419/ijet.v7i2.32.1558

[10.] Mohanty, M. D., Mohanty, B., &Mohanty, M. N. (2017). R-peak detection using theefficient technique for tachycardia detection. *2017 2nd International Conference on Man and Machine Interfacing (MAMI)*. https://doi.org/10.1109/mami.2017.8307877

[11.] Yano, T., Martins, E., & de Sousa, F. L. (2011). Most: A multi-objective search-based testing from EFSM. *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops*. https://doi.org/10.1109/icstw.2011.37

[12.] Roy, S., Chaulagain, D., &Bhusal, S. (2018). Static Analysis for Security Vetting of android apps. Lecture Notes in Computer Science, 375–404. https://doi.org/10.1007/978-3-030-04834-1_19

[13.] Shrivastava, G., & Kumar, P. (2019). Intent and permission modeling for privacy leakage detection in Android. Energy Systems. https://doi.org/10.1007/s12667-019-00359-7

[14.] Pan et al.(Pan, Y., Ge, X., Fang, C., & Fan, Y. (2020). A systematic literature review of Android malware detection using static analysis. IEEE Access, 8, 116363–116379. https://doi.org/10.1109/access.2020.3002842)

[15.] Jeon et al.(Jeon, G., Choi, M., Lee, S., Yi, J. H., & Cho, H. (2021). Automated multi-layered bytecode generation for preventing sensitive information leaks from Android Applications. IEEE Access, 9, 119578–119590. https://doi.org/10.1109/access.2021.3107601)

[16.] Paul et al. (Paul, N., Bhatt, A. J., Rizvi, S., & Shubhangi. (2022). Malware detection in Android apps using Static Analysis. Journal of Cases on Information Technology, 24(3), 1–25. https://doi.org/10.4018/jcit.20220701.oa6)