Mobile Malware Identification Technique

[1]**RAJANIKANTA SAHU,** *Gandhi Institute of Excellent Technocrats, Bhubaneswar, India*
[2]**SATYABRATA NAIK,** *Capital Engineering College, Bhubaneswar, Odisha, India*

Abstract -- Mobile phone usability has seen incredible innovation and growth. Many applications and games accessible for free on Google Play can lead to the horror of downloading malware software. As a result, users will need some computing power to run highly complex effective algorithms for mobile intrusion detection discovery, which will be impossible to implement on mobile devices. As a result, a robust platform such as the cloud appears to be required to defend users from threats and various security challenges. Cloud computing has taken the world by storm, with a variety of cloud-based intrusion detection systems (IDS) that can improve Smartphone security and performance.

An examination of the words connected to mobile malware tactics, classes, and techniques is presented in this review paper.

Index Terms-- Malware techniques, Mobile cloud computing, Mobile malware detection, Intrusion Detection System.

I. INTRODUCTION

Mobile phones are not as safe as they appear. Mobile devices face a number of serious vulnerabilities and security threats. According to the International Data Corporation (IDC), phone companies delivered a total of 341.1 million smart phones around the world in the second quarter of 2018 (2Q18), and the Android platform has risen to the top of the market, accounting for 84.8% of the Smartphone market. [1] Because Android controls the majority of the market, malware continues to increase; thousands of new infections are released. The term malware is derived from the combination of the words malignant and software, and has come to signify any harmful software. It recognizes any code included, transmuted, or preoccupied from a software system with the intent of causing harm or subverting the system's intended operation. Malware is defined by its ability to replicate, execute itself, and corrupt the PC framework.

The current infection is capable of delivering user contact lists and other information, completely locking down the device, granting remote access to [3] criminals, sending SMS and MMS messages, and so on. Malware operations have almost exclusively targeted Google Play store consumers, according to McAfee.

Figure 1 shows everything from the earliest banking Trojan on Google Play, nicknamed Droid09, to the most recent advertisement click misrepresentation/Bit coin-mining apps. [4] The Google Play store has been targeted..

There are numerous malware samples, as shown in Figure 2 [4], which depicts the rise of the malware industry from Q3 of 2016 to Q2 of 2018. Both Google and Apple are concerned about security. [5]

Fig. 1. Threats targeting Google play in 2016.

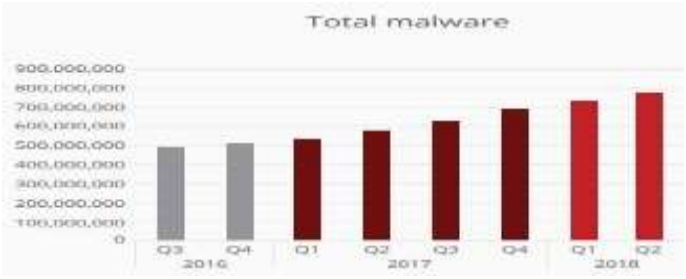

Fig. 2. Total malware samples from 2016 – 2018.

The sections that follow are grouped as follows: The second section provides an overview of mobile malware, including scientific classification and security risks. The third section summarizes the performance assessment measures and criteria for mobile malware detection. The fourth section organizes cloud-based IDS by architecture and type. The fifth section discusses mobile malware detection strategies, technologies, types, classification approaches, static techniques, and dynamic techniques, as well as a full comparison of existing mobile malware detection methods based on various characteristics. The last section concludes with a conclusion and recommendations for future work.

A SUMMARY OF MOBILE MALWARE

The type of virus affects a device's performance. Malware comes in a variety of shapes and sizes.

There are two classes. Adware, cookies, trojan horse, spam, mobile spyware, and mobile botent are examples of network-based malware (virus and worm). Table I displays a collection of malware and their characteristics. Malicious software has financial motivations, allowing attackers to make money. Some programmers send SMSs without the client's details, which then appears on the client's bill. Moua-bad, on the other hand, is a spyware that makes secret phone calls after locking the home screen. [6]

Table I. Taxonomy and examples of mobile malware with their behavior

Malware	**Platform**	**Cate-gory**	**Threats Behavior**
DroidKungFu	Android	Trojan horses	This malware introduces an indirect access in the android OS that enables programmers to increase full power over a client's cell phone.
Plankton	Android	Trojan horses	Take client program information and could get to a remote server to add considerably more malicious documents to the device. The malware would then be able to send client data to a remote server.
DroidLight			
DroidDream			
Privacy-A	IOS		
SPIsSaga	Symbian		
Geinimi	Android	Trojan horses	Geinimi conveys a user's location to hackers. Android.Pjapps send messages to premium rate numbers, which thusly pays to the makers of the Trojan.
Android.Pjapps			
shurufa	Symbian		
Zitmo	Black-berry	Trojan horses	Hack clients' financial balances by cell phone.
	Symbian		
InSpirit	Symbian	Worm	Take client's qualifications, for example, account points of interest by furtively tuning in to instant messages, catching key logging and so forth.
Ikee.B	IOS		
Fake-Player	Android	Trojan horses	Generate premium rate calls and SMS.
Floker	Symbian		
Android.Counterclank	Android	Trojan horses	for Android devices that steals information.
Android.Acnetdoor			open a back door on the compromised device.
Android.Answerbot			
NotCompatible	Android	Mobile botnet	Various interconnected bargained mobile devices execute undertakings.
Zitmo	Blackberry	Mobile Spyware	Keep an eye on any activities of mobile phone clients.
UAPush	IOS	Adware	Packaged with obscure programming through spring up advertisements or by some other way to do business promotions without the authorization of clients.

II. MALWARE TYPES

For making malware, aggressors use diverse courses going from clear standard systems that embedding's an outstanding piece of codes into a program document, to complex ones that use refined calculation to make obfuscated and polymorphic malware. [7]

A. Ordinary Malware (Static)

This kind of malware can be distinguished effectively by separating some unique characteristics which famous of a signature.

B. Polymorphic Malware

There is variable malware in which sentence structures of mal-code change in each time of infection, however the semantic proceed as before with no critical change at all.

C. Obfuscated Malware (Dynamic)

Incorporate polymorphic and transformative malware, in which the first code changed into a shape that is practically the equivalent however is substantially harder to be comprehended.

D. Encryption Malware

Encryption procedures are the most generally perceived strategies used in polymorphic malware.

III. DETECTION PERFORMANCE EVALUATION

With the end goal to secure mobile phones and resist threats mentioned in Section 2.

There are prerequisites should satisfy by the measures for evaluating detection performance as an Indication of good or bad performance. Measures allude to the evaluation metrics that are utilized to assess the performance of a detection method. By utilizing these indications and measures a high accomplish of performance can be reached and achieve design expectation.

A. *Indication of Successful Malware Detection*

Various indications ought to evaluate the performance of a dynamic mobile malware detection method. Thus, various underlined indications proposed as depicted

- Detection accuracy is a fundamental rule to assess the method. The higher the percentage, the better the method performs. In section B a number of measures are proposed to measure detection accuracy. [8]
- Privacy preservation. Transfer data from a mobile device to a third party could cause information leakage or malicious data manipulation. Be that as it may, safeguarding client protection and information security is basic in this kind of method. [9]
- Ability to recognize obscure applications. There is need to recognize obscure applications and zero-day attacks. [10]
- Real-time detection support. It can gather and breaking down application runtime information constantly without affecting application execution. A perfect arrangement is that distinguishing proof and examination are performed inside the mobile phones. Recognition ought to be quick and effective.
- Ability to switch between available algorithms to enhance detection performance [8]. Along these lines, appropriately picking a classification algorithm could influence the performance of detection.

B. *Evaluation Metrics*

There are numerous estimates that can be utilized to assess detection accuracy, as mentioned in Table II. [11]
Table III presents four essential classification measures in wording of the connection among malicious and benign status of the application. [12]

Table III. The summary of four basic measures for evaluation.

		Prediction	
		Malicious	**Benign**
Reality	**Malicious**	TRUE NEGATIVE	FALSE POSITIVE
	Benign	FALSE NEGATIVE	TRUE POSITIVE

Table II. Measures used for detection performance evaluation

Measures	**Explanation**
True Positive (TP)	Malicious programs 100% effectively distinguished as malicious program.
False Positive (FP)	Benign programs inaccurately distinguished as malicious.
True Negative (TN)	Benign programs 100% effectively distinguished as benign.
False Negative (FN)	Malicious programs inaccurately distinguished as benign.
Recall: True Positive Rate (TPR)	$TPR = \frac{TP}{TP+FN}$
True Negative Rate (INR)	$TNR = \frac{TP}{TN+FP} = 1 - FPR$
False Positive Rate (FPR)	$FPR = \frac{FP}{FP+TN}$, i.e., false alarm rate.
Precision: Positive Prediction	$P = \frac{TP}{TP+FP}$
F-score (F-measure)	F-score = F-measure= $(1 + \alpha^2) \frac{P \times TPR}{\alpha^2(P+TPR)}$,α is a predefined parameter.
Accuracy =	$ACC = \frac{TP+TN}{TP+FP+TN+FN}$
Receiver Operating Characteristic (ROC)	Calculated by FPR and TPR as x- and y-axes, as determined the trade-offs between True Positive and False Positive.
Area Under the	AUC = $\int_{-\infty}^{+\infty} TPR(t)\, FPR(t)dt$.

IV. MOBILE MALWARE DETECTION

A. *Malware Detection Techniques*

In the mobile environment, there are two classifications: Anomaly-Based Detection and Signature-Based Detection. Figure 3. Indicates distinctive approaches, which go under these techniques. A particular analysis or approach of both the techniques is dictated by how specific techniques accumulate data to recognize and detect malware.

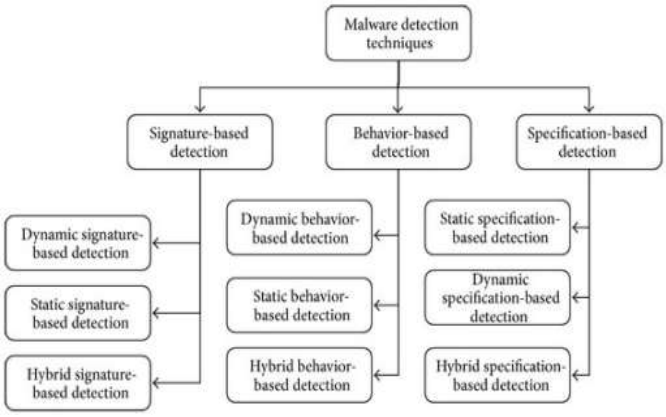

Fig. 3. Malware Detection Technique [13]

1) Signature-based techniques: The pernicious practices of know malware are caught as their signatures. When one of its signatures is perceived, the malware is recognized. [14]

2) Anomaly-based techniques (behavior-based): The typical system conduct is displayed first. At that point, the malware is identified at whatever point the system conduct goes amiss from the displayed ordinary conduct. [15]

3) Heuristic based techniques (specification-based): Artificial intelligence (AI), signature and anomaly-based techniques to upgrade their proficiency. [16]

B. Intrusion Detection Methods

All malware scanners, essentially, utilize signature and anomaly -based techniques for perceiving personalities of programs.

1) Dynamic methods: Utilization run-time data of a malware, when it is executed in a memory.

2) Static methods: Those are finished by extricating features from static malware when it is in a disk.

3) Hybrid methods: Utilization mix of dynamic and static methods. [17]

C. Malware Detection Types

1) Host-based intrusion detection system (HIDS): monitor dynamic conduct and condition of particular PC framework to check whether there are any inner or outside activities swindle the framework approach. [18]

2) Network-based intrusion detection system (NIDS): Used to sniff every one of the parcels on network nodes for examination. In this create a lone sniffer module set in every framework segment to screen traffic in that fragment. Interestingly dispersed system based interruption location framework has different modules put in each hub to screen movement in those nodes or hubs. [19]

D. Mobile Malware Detection Methods

It is a challenge to perceive malware dynamically in mobile devices especially when malware originators use encryption algorithms. the detection techniques of mobile malware advance so fast. [11]

1) Classification algorithms: Go for ordering obscure samples with legitimate labels, for example malicious or benign. They fill in as the most fundamental piece of malware detection together with the features. The most well-known technique utilized in classification is machine learning, went with data mining methods. Statistical methods and programming the most part utilize for data mining. To discover patterns of features, which can be connected into machine learning to how build classification models. Most classification algorithms fall into the extent of machine learning. [11] In Table IV, a briefly introduction about the most popular machine learning-based classification algorithms.

Table IV Frequently used classification algorithms

Algorithms	**Advantages**	**Disadvantages**
Naive Bayes	Very fast and simple	Require presumption of shared freedom of features
K Nearest Neighbors	High percentage of accuracy	Coast very big load to un determined data set
Decision Trees	Ability to deal with undetermined data set or features whatever size of data	Difficulty to control the process
Random Forest	Ability to deal with very huge data like multi-dimensional data set	Can reach to bad percentage of accuracy
K-means	Ability to control the process	The value of K need to be predefined depend on the running process
AdaBoosting	Reach High percentage of precision	Sensitive to undefined dataset
Logistic regression	high speed of processing	Cannot control huge data like dimensional data
C4.5 (=J48)	Easy to understand produced rues of process	Low capacity

Choosing a suitable classification algorithm is astoundingly imperative since it impacts discovery precision and execution. In case having a small training set, use Naive Bayes or K-implies algorithm. Random Forest could be a proper algorithm with many types of features needed to be into consideration. The algorithms, such as K-means, K Nearest Neighbors (KNN), and Random Forest, need to predefine a few

parameters for recognizing malware. In addition, the estimations of parameters incredibly affect detection performance.

2) Static techniques: Depend on the analysis of the detecting source code of exciting application to classifying accordingly without need to executing it. As it is a very fast and inexpensive techniques of detecting any bad code segment or malicious behave of any segment of code.

Those techniques are classified into classes according to what technique is used for analyzing the source code in order to detect any malicious code segment into target application. This section divided the static detection methods into three types. In each type the most current literature work that used this specific technique are discussed.

a) Signature based analysis

Min et al. [19] proposed an analytical system for android which can automatically identify malicious code segment, collect malware and generate signatures for applications.by using permission recursion technique and class association, this analytical system success in detecting a Zero-day malware which is a new malware that current anti-virus systems cannot detect. The system is being used to detect 2,475 malware samples from 102 families, with 327 zero-day malware samples belongs to six different families.

According to [7] a program classifies a malware if its signature matched the exciting signatures that are available in anti-virus program by extracting the semantic patterns and creates a unique signature. It considers a very fast method for detecting malware, however, it needs immediate update of malware signature to keep up with the revolution in the industry of malware.

Halilovic et al. [20] proposed a system that access the log file of user then compare it according to rule-sets. In case of no matching item, normal action is done. Otherwise a new malware signature is detected and there is system alarm.

Alam et al. [3] build signature-based model using a Random Forest algorithm based on Weka, which provide a database contain samples of malicious features to work into it. Un fortunately the detection doesn't support the real-time detection but the accuracy rate reach to 80%.

b) Permission based analysis

According to [6] permissions include the following requested and required permissions. But android permissions are classified into four types: signature, normal, dangerous and signature or system. So, in order to known the malicious application, there is need to check weather an application requires a dangerous permission access or not.

Zarni Aung et al. [21] implement a framework, as K-Means Clustering Algorithm is chosen to determine if the target application is malicious or normal application. In order to develop this model, they have extracted various permission from several downloaded applications from android markets. This module has been evaluating by using the Area Under ROC Curve (AUC). This system achieves on testing on dataset1 a true positive rate 90.72% and false positive rate reach to 9.27% otherwise on testing on dataset2 a true positive rate 85.05% and false positive rate reach to 4.94%.

Shuang et al. [22] Droid Detective, an android malware detection system was proposed to enhance the Security of mobile devices. It is offline tools, which depend on permissions analysis combinations in order to detect any mobile malware. This tool developed by using K-maps algorithm. Droid Detective achieve false positive rate = 12.47%, false negative rate= 16.43% and with positive rate = 87.53%.

Xing Liu and Jiqiang Liu. [17] proposed a scheme by extract several Permissions from .APK files: Requested Permissions, Request Permission Pairs, Used Permissions, Used Permission Pairs. This schema TPR IS 80.5%, FPR 0.5% and with accuracy 98,6%.

Yerima et al. [12] a Genome Project data classifier for android application by using *Bayesian* classifier based on static code analysis. A feature extracted from *.apk* files like: API calls, Linux system commands and permissions so as to results announced preferable discovery rates over signature-based antivirus.

c) Virtual machine analysis

In mobile applications a virtual machine is needed to examine the byte code of a particular application.by analyzing the byte code, there are a lot of aspects that can be examined like: application behavior, data flow of application and analysis control. all of these functionalities are helpful into detecting malicious behavior performed by any type of malware apps. The limitation of virtual machine is that analysis performed at the instruction level and consume from the storage and power of the apps.

DroidAPIMiner is presents by Y. Aafer et al. [8]as an analytical virtual machine to examine android app through tracking the API calls.

3) Dynamic techniques: depend on the analysis of the detecting application during the it's execution time in order to monitor dynamic behavior. This technique is done in runtime. This technique overcome the limitation of static analysis like obfuscation. One of its advantages that there is a large scale for analysis regardless the static analysis. Those techniques are classified into classes according to what techniques is used for analyzing the app. This section divided the static detection methods into three types: anomaly-based detection, taint analysis detection, and emulation-based detection.
In each type the most current literature work that used this specific technique are discussed.

a) Anomaly based analysis

One of the most popular approach of the behavior-based detection is the anomaly-based detection, as Behavior based methods concentrate on the behavior of the system from outside by executing it. If there is any malicious behavior after

observing the behavior of system, it can be identified as malware. There are several types of behavior-based detection exists, depending on the way that the approach keeps tracking of different parameters and the status of different components of the target device.

CrowDroid [8] is an anomaly-based detection tool, used for malware detection in android mobiles by analyzing the system call's logs in order to know the behavior of the target device while AntiMalDroid depending on analyzing the behavior of system, then generate a signature of system's malware behavior. For IOS devices, there are popular anomaly-based detection tools like SMS Profiler and iDMA both of them keep tracking of illegitimate usage of system services.

Radoglou et al. [23] proposed a lightweight IDS for detecting malicious behavior for android devices which used a very powerful multi-layer perception (MLP) neural network. This system consists of three components: information source, analysis engine, response. There is a machine learning algorithm for detecting unknown threads with accuracy reaches to 81, 39% and detection rate reaches to 85,02%. The main goal of system is to achieve very high rates of malicious behavior detection with small rates of false alarms. The detection in this system achieved by monitoring the Net Flows, then IDS, which has a strong Python backend analyzing the network traffic, and matched it with (MLP) neural network. If there is matching an alert is fired for detecting an intrusion.

Mohata et al. [3] introduced an anomaly-based system, which consist of two stages: first constructs signatures for the API calls of target device then train a classifier using a support vector machines (SVMs) in order to distinguish between a malicious programs from benign program.

According to [20], a defensive program used to examine files before user download it. First all information about file is entered by user through a web service, then a string matching and file information been processed by cloud server for comparison. The detecting process performed in cloud service for defining any intrusions.

Hua et al. [24] introduced a framework for an anomaly-based intrusion detecting IDS to monitor all information about android mobile devices then analysis all of it by a Naive Bayes Algorithm to classify the collected data as normal or malicious.

b) Taint analysis

Taint, one of the most famous terms. To taint user data, it means that to insert some kind of tag or label for each object of the user data. The tag allows us to track the influence of the tainted object along the execution of the program. It is a technique to mark the most important information to track with an identifier called "Taint".

According to [7], TaintDroid is frameworks to track the flow of sensitive data throw third party apps at the android platform. TaintDroid has two concepts of modification standard android system and manual tagging.

On the other hand, according to [25] AndroTaint is an efficient android malware detection framework that use dynamic taint analysis without need for android system modification and use automatic tagging. This tool depends on the data set of malwares. This approach using dynamic taint analysis. All leakage sensitive information like Call-Logs, Contacts, existing SMS, Email, SD-card contents and GPS location co-ordinates identified as taint source then taint marking stars with help of automatic tagging. The statistics of AndroTaint is 90% (malicious Apps: 80%, benign Apps: 90%, aggressive Apps: 94% and 95% risky Apps: 95%). However, both tools TaintDroid and AndroTaint explicit flow analysis without covering the control flow analysis.

c) Emulation based analysis

According to [26], Mobile sandbox to test untrusted users, websites or third parties without risking harm to host machine or OS. This tool is a combination of static and dynamic analysis. The static analysis used for APK file like user permission and manifest.xml file to identify any malicious code. Dynamic part used to check the network flow and native calls to standing the behavior of the suspicious application.

On the other hand, there is DroidScope [8] is an emulation tool used to dynamically analyzing the suspicion application based on virtual machine. It is monitoring the system by being out of execution environment.

4) Comparison of mobile malware detection methods

# Ref	IDS type	Techniques	Positioning	Algorithms	Evaluation Measure Type and Value	Pros & Cons
Min et al. [25]	HIDS	Signature-based	On each host	-	detect 2,475 malware samples.	success in detecting a Zero-day malware but suffer from recursion and repetition.
Guo et al. [3]	HIDS	Signature-based	On each host	Naive Bayes algorithms	accuracy rate reach to 80%.	semantic patterns and creates a unique signature, but this detection cause time consumption.
Alam et al. [27]	HIDS	Signature-based	On each host	Random Forest algorithm	accuracy rate reach to 80%.	a new malware signature is detected and there is system alarm, but un- fortunately the detection doesn't support the real-time detection.
Zarni Aung et al. [21]	HIDS	Permission-based	On each host	K-Means Clustering Algorithm	This module achieves a true positive rate 90.72% and false positive rate reach to 9.27% on testing dataset1 otherwise on testing on dataset2 a true positive rate 85.05% and false positive rate reach to 4.94%.	This system determines if the target application is malicious or normal application, but doesn't detect zero days malwares.
Shuang et al. [22]	HIDS	Permission-based	On each host	K-maps algorithm	false positive rate = 12.47%,. false negative rate= 16.43% . positive rate = 87.53%.	It is offline tools which depend on permissions analysis combinations in order to detect any mobile malware, but doesn't detect some types of zero days malwares.
Xing Liu and Jiqiang Liu. [17]	HIDS	Permission-based	On each host	machine learning techniques	TPR = 80.5%,. FPR = 0.5% . accuracy = 98,6%.	detecting Android malicious applications by extracting several features from a large number of APK files: Requested Permissions, Request Permission Pairs, Used Permissions, Used Permission Pairs. But there is consumption of time.

Yerima et al. [12]	HIDS	Permission-based	On each host	Bayesian classifier	accuracy rate reach to 65-85%.	based on static code analysis. achieve high rates of accuracy than signature-based antivirus. But there is dynamic support analysis for any zero days malwares.
CrowDroid [8]	HIDS	Anomaly-based	On each host	K-Means Clustering Algorithm	-	analyzing the system call's logs in order to know the behavior of the target device while AntiMalDroid depending on analyzing the behavior of system, then generate a signature of system's malware behavior. keep tracking of illegitimate usage of system services. Which cause user less privacy.
Radoglou et al. [23]	NIDS	Anomaly-based	On network	(MLP) neural network	accuracy reaches to 81,39% and detection rate reaches to 85,02%.	achieve a very high percentage of accuracy with low percentage of false alarms. There is no saving of resources, which cause fast battery lost.
Hua et al. [24]	NIDS	Anomaly-based	On network	Naive Bayes Algorithm	-	high percentage of accuracy. However, coast high computation.
Y. Aafer et al. [8]	HIDS	Virtual machine	On each Virtual machine	-	accuracy rate reach to 70%.	an analytical virtual machine to examine android app through tracking the API calls
In paper [7]	NIDS	Taint - analysis	Between both host and network	machine learning techniques	-	track sensitive data throw third party apps at the android platform. But modify standard android system and used manual tagging.
In paper [25]	NIDS	Taint - analysis	Between both host and network	Bayesian classifier	accuracy rate reach to 90%.	explicit flow analysis without covering the control flow analysis.
In paper [26]	NIDS	Emulation-based	Between both the host and virtual machine	-	accuracy rate reach to 70-75%.	check the network flow and native calls to standing the behavior of the suspicious application. There is a risk not to known the running system malwares.
In paper [8]	NIDS	Emulation-based	Between both the host and virtual machine	-	accuracy rate reach to 75%.	dynamically analyzing the suspicion application based on virtual machine. But it monitoring the system by being out of execution environment.

V. Conclusion And Future Work

With the developing utilization of Smartphone, the quantity of assaults and dangers are additionally on increment. It is important to give security to end clients from dangers. In this paper, we represent a full picture about malware environment as discussing malware classes and techniques there are different techniques have been discussed and listed. Papers also mention Android malware detection types, methods, technologies and proposed techniques. In above section we have studied various algorithms, which restrict the detection of attacks.

In future, work a detailed study with the most effective tools to detect mobile real-time threads.

References

[1] https://www.idc.com/promo/smartphone-market-share/os [Accessed 6 September 2018]

[2] Joshi, P.; Jindal, C.; Chowkwale, M.; Shethia, R.; Shaikh, S. A. & Ved, D. Protego: A passive intrusion detection system for Android smartphones *Computing, Analytics and Security Trends (CAST), International Conference on*, **2016**, 232-237

[3] Mohata, V. B.; Dakhane, D. M. & Pardhi, R. L. Mobile Malware Detection Techniques *International Journal of Computer Science & Engineering Technology (IJCSET)*, **2013**, *4*, 2229-3345

[4] https://www.mcafee.com/enterprise/en-us/threat-center/mcafee labs/reports.html [Accessed 6 September 2018]

[5] Yan, P. & Yan, Z.A survey on dynamic mobile malware detection *Software Quality Journal, Springer*, **2017**, 1-29

[6] Malhotra, A. & Bajaj, K. A survey on various malware detection techniques on mobile platform *Int J Comput Appl*, **2016**, *139*, 15-20

[7] Saeed, I. A.; Selamat, A. & Abuagoub, A. M. A survey on malware and malware detection systems, *International Journal of Computer Applications, Foundation of Computer Science*, **2013**, *67*

[8] Amro, B. Malware detection techniques for mobile devices *arXiv preprint arXiv:1801.02837*, **2018**

[9] Gibler, C.; Crussell, J.; Erickson, J. & Chen, H. AndroidLeaks: Automatically Detecting Potential Privacy Leaks in Android Applications on a Large Scale. *Trust, Springer*, **2012**, *12*, 291-307

[10] Rastogi, V.; Chen, Y. & Jiang, X. Catch me if you can: Evaluating android anti-malware against transformation attacks, *IEEE Transactions on Information Forensics and Security, IEEE*, **2014**, *9*, 99-108

[11] Mujumdar, A.; Masiwal, G. & Meshram, D. B. Analysis of signature-based and behavior-based anti-malware approaches *signature*, **2013**, *2*

[12] Pehlivan, U.; Baltaci, N.; Acartürk, C. & Baykal, N. The analysis of feature selection methods and classification algorithms in permission-based Android malware detection *Computational Intelligence in Cyber Security (CICS), 2014 IEEE Symposium on*, **2014**, 1-8

[13] https://openi.nlm.nih.gov/detailedresult.php?img=PMC4138763_TSWJ2014-983901.005&req=4

[14] Papamartzivanos, D.; Damopoulos, D. & Kambourakis, G. A cloud-based architecture to crowdsource mobile app privacy leaks *Proceedings of the 18th Panhellenic Conference on Informatics*, **2014**, 1-6

[15] Ham, H.-S. & Choi, M.-J. Analysis of android malware detection performance using machine learning classifiers *ICT Convergence (ICTC), 2013 International Conference on*, **2013**, 490-495

[16] Ahmadi, M.; Biggio, B.; Arzt, S.; Ariu, D. & Giacinto, G. Detecting misuse of google cloud messaging in android badware *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices*, **2016**, 103-112

[17] Liu, X. & Liu, J. A two-layered permission-based Android malware detection scheme *Mobile cloud computing, services, and engineering (mobilecloud), 2014 2nd ieee international conference on*, **2014**, 142-148

[18] He, D.; Chan, S. & Guizani, M. Mobile application security: malware threats and defenses *IEEE Wireless Communications, IEEE*, **2015**, *22*, 138-144

[19] Zheng, M.; Sun, M. & Lui, J. C. Droid analytics: A signature based analytic system to collect, extract, analyze and associate android malware*Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, **2013**, 163-171

[20] Supriya Kamble, Leena Ragha and Puja Padiya. Comparative Study on Intrusion Detection Systems for Smartphones, *IOSR-JCE*, **2015**

[21] Aung, Z. & Zaw, W. Permission-based android malware detection *International Journal of Scientific & Technology Research*, **2013**, *2*, 228-234

[22] Liang, S. & Du, X. Permission-combination-based scheme for android mobile malware detection *Communications (ICC), 2014 IEEE International Conference on*, **2014**, 2301-2306

[23] Radoglou-Grammatikis, P. I. & Sarigiannidis, P. G. Flow anomaly-based intrusion detection system for Android mobile devices *Modern Circuits and Systems Technologies (MOCAST), 2017 6th International Conference on*, **2017**, 1-4

[24] Sarafarazahmad Momin. Automated Mobile Web Apps Testing Tool, IJSR, **2014**

[25] Shankar, V. G., Somani, G., Gaur, M. S., Laxmi, V., & Conti, M. (2017, January). AndroTaint: An efficient android malware detection framework using dynamic taint analysis. In *Asia Security and Privacy (ISEASP), 2017 ISEA* (pp. 1-13). IEEE

[26] Kaur, R.; Kumar, G. & Kumar, K. A comparative study of feature selection techniques for intrusion detection *Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on*, **2015**, 2120-2124

[27] Joy, R. & Ajith, A. SURVEY ON ANDROID MALWARE DETECTION METHODS USING STATIC AND DYNAMIC ANALYSIS *International Journal*, **2016**, *5*

[28] Shabtai, A. & Elovici, Y. Applying behavioral detection on android-based devices *Mobile Wireless Middleware, Operating Systems, and Applications, Springer*, **2010**, 235-249

[29] Narudin, F. A.; Feizollah, A.; Anuar, N. B. & Gani, A. Evaluation of machine learning classifiers for mobile malware detection *Soft Computing, Springer*, **2016**, *20*, 343-357

[30] Dalla Preda, M. & Maggi, F. Testing android malware detectors against code obfuscation: a systematization of knowledge and unified methodology *Journal of Computer Virology and Hacking Techniques, Springer*, **2017**, *13*, 209-232

[31] Hamed, T.; Ernst, J. B. & Kremer, S. C. A Survey and Taxonomy on Data and Pre-processing Techniques of Intrusion Detection Systems, Computer *and Network Security Essentials, Springer*, **2018**, 113-134

[32] Wang, X.; Yang, Y. & Zeng, Y. Accurate mobile malware detection and classification in the cloud *SpringerPlus, Nature Publishing Group*, **2015**, *4*, 583

[33] Hamed, T.; Ernst, J. B. & Kremer, S. C. A survey and taxonomy of classifiers of intrusion detection systems *Computer and network security essentials, Springer*, **2018**, 21-39

[34] Modi, C.; Patel, D.; Borisaniya, B.; Patel, H.; Patel, A. & Rajarajan, M. A survey of intrusion detection techniques in cloud *Journal of Network and Computer Applications, Elsevier*, **2013**, *36*, 42-57

[35] Feizollah, A.; Anuar, N. B.; Salleh, R. & Wahab, A. W. A. A review on feature selection in mobile malware detection *Digital Investigation, Elsevier*, **2015**, *13*, 22-37

[36] Liao, H.-J.; Lin, C.-H. R.; Lin, Y.-C. & Tung, K.-Y. Intrusion detection system: A comprehensive review *Journal of Network and Computer Applications, Elsevier*, **2013**, *36*, 16-24

[37] Aljawarneh, S.; Aldwairi, M. & Yassein, M. B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model *Journal of Computational Science, Elsevier*, **2018**, *25*, 152-160

[38] Roman, R.; Lopez, J. & Mambo, M. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges, *Future Generation Computer Systems, Elsevier*, **2018**, *78*, 680-698
[39] Moran, K.; Linares-Vásquez, M.; Bernal-Cárdenas, C.; Vendome, C. & Poshyvanyk, D. Automatically discovering, reporting and reproducing android application crashes *Software Testing, Verification and Validation (ICST), 2016 IEEE International Conference on*, **2016**, 33-44
[40] Mirzaei, N.; Garcia, J.; Bagheri, H.; Sadeghi, A. & Malek, S. Reducing combinatorics in gui testing of android applications, Software *Engineering (ICSE), 2016 IEEE/ACM 38th International Conference on*, **2016**, 559-570
[41] Fridman, L.; Weber, S.; Greenstadt, R. & Kam, M. Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location, *IEEE Systems Journal, IEEE*, **2017**, *11*, 513-521
[42] Elgendy, I. A.; El-kawkagy, M. & Keshk, A. Improving the performance of mobile applications using cloud computing *Informatics and Systems (INFOS), 2014 9th International Conference on*, **2014**, PDC-109, Added **to IEEE *Xplore*: 12 February 2015**
[43] Faruki, P.; Bharmal, A.; Laxmi, V.; Ganmoor, V.; Gaur, M. S.; Conti, M. & Rajarajan, M. Android security: a survey of issues, malware penetration, and defenses *IEEE communications surveys & tutorials, IEEE*, **2015**, *17*, 998-1022
[44] Suarez-Tangil, G.; Tapiador, J. E.; Peris-Lopez, P. & Ribagorda, A. Evolution, detection and analysis of malware for smart devices *IEEE Communications Surveys & Tutorials, IEEE*, **2014**, *16*, 961-987
[45] Walls, J. & Choo, K.-K. R. A review of free cloud-based anti-malware apps for android, *Trustcom/BigDataSE/ISPA, 2015 IEEE*, **2015**, *1*, 1053-1058
[46] Qureshi, S. S.; Ahmad, T.; Rafique, K. & others Mobile cloud computing as future for mobile applications-Implementation methods and challenging issues *Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on*, **2011**, 467-471
[47] Gao, J.; Bai, X.; Tsai, W.-T. & Uehara, T. Mobile application testing: a tutorial, *Computer, IEEE*, **2014**, *47*, 46-55
[48] Ghorbanian, M.; Shanmugam, B.; Narayansamy, G. & Idris, N. B. Signature-based hybrid Intrusion detection system (HIDS) for android devices *Business Engineering and Industrial Applications Colloquium (BEIAC), 2013 IEEE*, **2013**, 827-831
[49] Houmansadr, A.; Zonouz, S. A. & Berthier, R. A cloud-based intrusion detection and response system for mobile phones *Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on*, **2011**, 31-32
[50] Peiravian, N. & Zhu, X. Machine learning for android malware detection using permission and api calls *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, **2013**, 300-305
[51] Linares-Vásquez, M.; Vendome, C.; Luo, Q. & Poshyvanyk, D. How developers detect and fix performance bottlenecks in android apps *Software Maintenance and Evolution (ICSME), 2015 IEEE International Conference on*, **2015**, 352-361
[52] Peiravian, N. & Zhu, X. Machine learning for android malware detection using permission and api calls *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, **2013**, 300-305
[53] Zachariah, R.; Akash, K.; Yousef, M. S. & Chacko, A. M. Android malware detection a survey *Circuits and Systems (ICCS), 2017 IEEE International Conference on*, **2017**, 238-244
[54] Odusami, M., Abayomi-Alli, O., Misra, S., Shobayo, O., Damasevicius, R. and Maskeliunas, R., 2018, November. Android Malware Detection: A Survey. In *International Conference on Applied Informatics* (pp. 255-266). Springer, Cham.
[55] Penning, N.; Hoffman, M.; Nikolai, J. & Wang, Y. Mobile malware security challeges and cloud-based detection *Collaboration Technologies and Systems (CTS), 2014 International Conference on*, **2014**, 181-188
[56] Jadhav, S.; Dutia, S.; Calangutkar, K.; Oh, T.; Kim, Y. H. & Kim, J. N. Cloud-based Android botnet malware detection system, *Advanced Communication Technology (ICACT), 2015 17th International Conference on*, **2015**, 347-352
[57] Li, X.; Liu, J.; Huo, Y.; Zhang, R. & Yao, Y. An android malware detection method based on androidmanifest file *Cloud Computing and Intelligence Systems (CCIS), 2016 4th International Conference on*, **2016**, 239-243
[58] Silakari, S.; Chourasia, U. & others, Malware Detection Techniques in Cloud Computing Infrastructure using ACMPSO-k means *International Journal of Computer Science and Information Security, LJS Publishing*, **2016**, *14*, 29
[59] Aulakh, J. K.; Sharma, S. & Arora, M. Mobile Cloud Computing Security Issues: Overview *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume*, **2014**, *3*
[60] Yi, S.; Li, C. & Li, Q. A survey of fog computing: concepts, applications and issues, *Proceedings of the 2015 Workshop on Mobile Big Data*, **2015**, 37-42
[61] Hussain, A.; Razak, A. & Mkpojiogu, E. The perceived usability of automated testing tools for mobile applications, *Journal of Engineering Science and Technology*, ***2017***, *12*, 89-97
[62] Gai, K.; Qiu, M.; Tao, L. & Zhu, Y. Intrusion detection techniques for mobile cloud computing in heterogeneous 5G *Security and Communication Networks, Wiley Online Library*, **2016**, *9*, 3049-3058
[63] Perez, M. & Kumar, S. A Quick Survey on Cloud Computing and Associated Security, Mobility and IoT Issues, *Journal of Computer and Communications, Scientific Research Publishing*, **2017**, *5*, 80
[64] Sankaranarayanan, S. & Murugaboopathi, G. Secure Intrusion Detection System in Mobile Ad Hoc Networks Using RSA Algorithm, *Recent Trends and Challenges in Computational Models (ICRTCCM), 2017 Second International Conference on*, **2017**, 354-357
[65] Saadi, C. & Chaoui, H. Security Analysis Using IDs Based on Mobile Agents and Data Mining Algorithms, IJCSIT*) International Journal of Computer Science and Information Technologies*, **2015**, *6*, 597-60
[66] Khune, R. S. & Thangakumar, J. A cloud-based intrusion detection system for Android smartphones, *Radar, Communication and Computing (ICRCC), 2012 International Conference on*, **2012**, 180-184
[67] Anirudha A. Kolpyakwar and Prof. Pragati Patil. Intrusion Detection System for Android Smartphone in Cloud Environment, IJARCCE, **2015**
[68] Oyeleye Christopher, A.; Daramola Comfort, Y. & Akinpelu James, A. Mob-AIDS: An Intrusion Detection System for the Android Mobile Enterprise *IJCSI International Journal of Computer Science Issues*, **2014**, *11*
[69] Mohini, T.; Kumar, S. A. & Nitesh, G. Review on Android and smartphone security, *Research Journal of Computer and Information Technology Sciences*, **2013**, *2320*, 6527
[70] Arora, S.; Bawa, R. & Student, M. A review on Intrusion Detection System to Protect Cloud Data, *International Journal of Innovations E Advancement in Computer Science*, **2014**, *3*, 30-34
[71] Omar, M. & Dawson, M. Research in progress-defending android smartphones from malware attacks *Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on*, **2013**, 288-292
[72] Halilovic, M. & Subasi, A. Intrusion Detection on Smartphones, *arXiv preprint arXiv:1211.6610*, **2012**
[73] Song, H.; Ryoo, S. & Kim, J. H. An integrated test automation framework for testing on heterogeneous mobile platforms *Software and Network Engineering (SSNE), 2011 First ACIS International Symposium on*, **2011**, 141-145
[74]https://archive.ics.uci.edu/ml/datasets/Detect+Malacious+Executable(Anti Virus) [21 September 2018]
[75] https://cloud.google.com/solutions/mobile/mobile-app-backend-services [22 October 2018]
[76] https://source.android.com/security/ [30 October 2018]
[77] http://www.softwaretestinghelp.com/best-mobile-testing-tools/ [22 November 2018]
[78] https://forensics.spreitzenbarth.de/android-malware/ [25 November 2018]
[79] Tuvell, G. & Venugopal, D. Malware detection system and method for mobile platforms *Google Patents*, **2017**
[80] Kene, S. G. & Theng, D. P. A review on intrusion detection techniques for cloud computing and security challenges, *Electronics and Communication Systems (ICECS), 2015 2nd International Conference on*, **2015**, 227-232
[81] Shah, B. & Trivedi, B. H. Improving Performance of Mobile Agent Based Intrusion Detection System, *Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on*, **2015**, 425-430
[82] Vasilomanolakis, E.; Karuppayah, S.; Mühlhäuser, M. & Fischer, M. Taxonomy and survey of collaborative intrusion detection, *ACM Computing Surveys (CSUR), ACM*, **2015**, *47*, 55
[83] Ms. Diksha Kale, Dr. Sudhir Sawarkar and Prof. Vijay Bhosale, Revised Approach for Smartphone Security Using Cloud and Android Applications, International Journal of Advanced Research in Computer Engineering & Technology IJARCET, **2015**.
[84] Nandasana, P.; Kumar, R.; Shinde, P.; Dhyani, A. & Parte, R. Cloud Based Intrusion Detection System, *International Journal of Scientific and Research Publications, Citeseer*, **2015**, 327