Protecting Android based applications from malware affected through SMS messages

J. Sasi Bhanu[1], J. K. R. Sastry[2], T. Chandrasekhara Reddy[3]
[1]Department of Computer Science and Engineering, CMR Institute of Technology, Vaddeswaram, India
[2]Department of Electronics and Computer Engineering, KLEF Deemed to be University, Vaddeswaram, India
[3]Department of Computer Science and Engineering, KLEF Deemed to be University, Vaddeswaram, India

Article Info

Article history:

Received Dec 31, 2020
Revised Apr 9, 2021
Accepted Apr 12, 2021

Keywords:

Android
Malware
Malware architecture
Malware detection
Messaging systems

ABSTRACT

Users use Android-based applications for communicating through emailing, text messaging, and transmission of audio and video objects. The attackers manipulate the email, text, videos, or audio so that users' receipt of the messages causes malware through their handheld devices. A runtime routine is invoked, which causes damage to the local resources of the mobile phone. The manipulation of the messages is done using different signatures, making it difficult to recognize the same using a single approach. Multiple approaches are sometimes required to detect different signature-based incoming messages. Choosing a method that suits the signature of the incoming message is the key. Malware can also enter at the time of installing third-party apps, clicking on the links provided in the messages, installing and invoking the malware in the background. Many issues are involved in dealing with malware detecting, prevention, and curing. A comprehensive architecture is required to deal with every aspect of dealing malware. In this paper, a comprehensive architecture is presented that considers malware's issue, especially concerning malware affected through short message service (SMS) messages operated under the Android operating system. The disection of the SMS messages have been implemented and 99% accuracy has been achieved.

Corresponding Author:

J. K. R. Sastry
Department of Electrical and Computer Engineering
Koneru Lakshmaih Education Foundation
Vaddeswaram, Guntur District 522502, India
Email: drsastry@kluniversity.in

1. INTRODUCTION

Smartphones are being used for many purposes, including playing games, online information access, dealing with emails and messages, surfing the net, and making commercial transactions. The transactions carried by the users involve personal information stored in the address book and within short message service (SMS) messages, leading to many security concerns. Many smartphones are being released today and have become a target for attackers to distribute malware and perform malicious attacks. 80% of the smartphones are Android-based.

Malware is a malicious code that can do different damages to the handheld devices, such as writing a message, stopping a running program, and modifying a file. Also, malware can be triggered periodically or lie dormant undetected until some event triggers the code to activate. The attacking systems are further classified as Trojans, bots, viruses, backdoor, worms, and rootkits. Android is an operating system that runs on most smartphones. Android powers everything from 5G phones to stunning tablets.

Android provides security in terms of permissions and priorities. Users have no idea of what happens when permissions are granted, and the priority of execution of the applications is fixed. Sometimes providing some permission may lead to access of the application by the malware software. The application developers have no access to these permissions, which is surely one of the security breaches existing exposed by the android operating system. These days users are spending 80% of the time doing messaging, due to which reason, the messaging applications have become the target of the attackers. Attackers manipulate the message by taking control of the message while they are in transmission to the messaging server situated at a far location.

The attackers generally insert uniform resource locator (URL) links within the messages. When the users click on the link, it will lead to downloading the small programs and then installed and executed. The malware programs are made intelligent to find the complete operating environment and then find the critical operations that, when initiated, damage the mobile phone system. The SMS is built with malicious links or URLs which can easily connect to the website with that link. Attackers use a variety of physical and virtual means to spread malware that affects devices. Malicious programs can be delivered through a universal serial bus (USB)-based system or spread over the internet through downloads, which automatically download malicious programs to systems without the user's approval or knowledge.

Sophisticated malware attacks often feature a command and control server that allows attackers to communicate with the infected systems, infiltrate data, and remotely control the compromised device. The practice in which an attacker sends emails purporting to be from reputable companies to induce individuals to reveal personal information is called phishing when a similar type of attack is done using SMS. It is known as smishing. In smishing, a message containing a computer address is distributed by the aggressor to the person in question to diverting the user to a website that collects the data about the users and creates a record having the data collected from the user. The data is then used for checking and deriving the most confidential information through brute force methods. Sometimes programs are downloaded as pop-up programs, and the same is executed as background processes when allowed by the users. Users are always hard-pressed, asking the users to enable the pop-ups to display the page they are looking at promptly.

Sometimes, the users are prompted to subscribe to the site to better use the facility, collect important information about the registrations' names, and use the same for attacking. After registration is done, the users are displayed with websites provided with the contact information that the user can use to either send an email or make a call. The attacker collects details in authentication and then uses it to log in to different sites, especially to the sites related to financial and e-commerce transactions. The malicious websites lure the users in discounts, promotional codes, subscriptions, and coupons for getting discounting.

There are many such ways of attacking mobile applications, as annunciated in the literature. There is a continuous requirement to monitor ongoing applications' behavior and find if any malware is operating and mitigating the same if it exists. They're all also continuous requirements to find if any malware is incoming. If found, appropriate actions are to be taken. Malware can be injected through emails, web pages, SMS messages, and other communication means. Many methods have also been proposed in the literature for detecting malware entering through different processes. Many working situations lead to malware, and a different method needs to be considered for detecting and mitigating malware.

Malware is caused through different mechanisms especially inflicted through SMS messages. The signature of the Messages that cause the malware varies greatly. A single method cannot identify all the signatures of the SMS messages that inflict the malware. Malware is triggered through emails, web surfing, and when third-party APPS are installed. A comprehensive malware management model is required that caters to all types of SMS signatures, methods through which the malware is inflicted, how malware is mitigated, and how counteractions are taken when incidences of the malware are traced.

A comprehensive architecture has to be investigated and implemented as another APP that catches hold of every message by negotiating with the Android operating system, inspecting the same for the existence of the malware, and mitigating the same if available. The detection is to be done by subjecting the incoming message to different machine learning models since no single model will recognize the different signatures of the images that indicate the malware presence. If malware is not traced, then the message is released back.

2. RELATED WORK

Belaoued *et al.* [1] have proposed detecting the malware based on detecting the program file headers as there will be a change in the headers when code changes occur. They have used a hybrid filter wrapper to extract the characteristics/features that are most relevant and match the characteristics of the data. The polymorphic and metamorphic techniques are mostly based on pattern matching. Each variant is focused on making changes to the code using a different pattern, as presented by Bruschi *et al.* [2]. Ramu [3] have presented a survey on how mobile malware has been evolved and how the malware is detected, and how the mobile system is protected when attacking is done using malware.

Ham and Choi [4] have proposed a new feature set and used the same to detect malware by inputting the feature set to machine learning classifiers. The malware application also hides any acknowledgments transmitted by the telecom operator. Thus, the malware application can cause a series of malicious transactions affecting the users who operate using the messages. Hamandi *et al.* [5] have demonstrated the working of such an application. Aung and Zaw [6] have proposed a framework to detect malware within the apps. The framework is based on machine learning-based malware detection. The system is built based on the apps' permissions and the events generated by those applications.

Qu and Hughes [7] have presented that aggregating many signatures and using the same to detect the malware will be quite helpful, especially in reducing the signature database. Blokhin *et al.* [8] have presented a code-sharing analysis method that focuses on partitioning the malware call logs into system call subsequence by identifying the logs' locations where the set of saved instruction pointers on the program call stack changes significantly.

Shaerpour *et al.* [9] have surveyed different techniques prosed in the literature for detecting malware. The techniques surveyed include malware detection using host-based frameworks, static analysis of executable, feature analysis, and behavioral patterns. Park *et al.* [10] have presented a method that focuses on finding the applications' similarity with the existing malware that runs the Android platform.

Yoon *et al.* [11] have analyzed financial transactions caused using SMS, and the users start attacking using the SMS data. Hamandi *et al.* [12] have presented different kinds of attacks on Android's messaging framework. Wang *et al.* [13] have proposed a method called Droid Chain to fight against malware variants and zero-day malware. To counter the system-call injection method, Naval *et al.* [14] have proposed a method that characterizes program semantics using asymptotic equipartition.

Zhang and Jin [15] proposed an enhanced malware detection method that combines both static analysis and ensemble techniques to improve malware detection accuracy. S. Chen *et al.* [16] have proposed a framework called StroDroid built with streamlined machine learning caused by enhanced features observed through a large data set collection.

Demontis *et al.* [17] proposed “Drebin,” a malware detector that implements corresponding evasion attacks. The detector is developed using a simple, scalable learning paradigm that mitigates the impact of evasion attacks. The method slightly worsens the detection rate in the absence of the attack. Cerbo *et al.* [18] have presented a methodology for carrying forensics analysis to detect malicious malware-based applications.

Grisham *et al.* [19] have used recurrent neural networks to identify the infected attachments and then carry the social network analysis for finding the key hackers disseminating from mobile devices. Amro [20] have analyzed different malware detection techniques that suit mobile operating systems like Android and iOS. Azar *et al.* [21] have presented a method, "analytics," that focuses on extracting static features of any binary file to distinguish malware. Arif *et al.* [22] have presented the way malware can enter using audio exploitation. Classification of audio files will help in developing a database required to detect malware entering through audios.

Arif *et al.* [22] have adopted a method that helps to analyze the audios and determine the static and dynamic characteristic of the audios, which are then used to find the system calls and permissions required for audio exploitation and the calls are applied on to different audio systems to the extent that damaging can be carried on different audios. Utku *et al.* [23] have developed a decision tree-based system using C4.5 and Hoeffding tree for detecting the malware coming into the mobile phones that operate on the Android operating system.

Iqbal and Zulkernine [24] dealt with real-time monitoring of the devices as Android allows low-level information to any apps introduced by third-party service providers. They have presented a real-time monitoring and detection technique called SpyDroid, which employs multiple detectors developed by third parties to detect the malware and allows for continuous monitoring of the devices in real-time. A number of contributions have been made to secure data while in a cloud and out of cloud which is not coupled with sprawling malware [25]-[32]. Panda and Tripathy [33] have presented a morphological malware detection method where malware signatures keep changing from time to time. A neural network-based learning method has been presented, which are connected with malware detection and prevention [34]-[38].

Many methods are used for detecting the malware. The methods used are increasing day after day. No scientific mechanism exists that can be used to classify the class to which the SMS message belongs which is the most important consideration that must be affected before even learning the machine learning method that should be used to detect the malware.

3. FUNCTIONAL REQUIREMENTS OF A MALWARE DETECTION, PREVENTION AND MITIGATION SYSTEM

A comprehensive approach to handle malware is to recognize all aspects in total and come out with a requirement list with detailed specification so that same can be considered to build into an architecture that that include all the components, the interaction between the components, the implementation of which can be done any platform. The following requirements have been identified through a detailed review of the literature and discussions with policemen who work in crime identification and eradication.

3.1. Requirement-1

Many malware detection methods focus on checking the code and the file structure used to store the binary in the file. Some malware detection technique focuses on learning the incoming messages through different machine learning techniques such as decision tree and neural networks. Some methods focus on feature extraction and similarity checking. These methods work perfectly in some contexts. As the contexts change, the methods become effective. A framework that caters to possible methods that can be used for detecting malware is required.

3.2. Requirement-2

Thus, there should be a continuous monitoring system that detects the existing malware and then quarantines the apps found to be malware while keeping checking instream mechanisms that can produce malware through SMS messages and other means. Thus, a mechanism is required that keeps monitoring the malware's existence continuously. There should be a method that keeps detecting the incoming messages to find any malware in them. Probability-based models are more realistic when it comes to continuous information coming through SMS messages.

3.3. Requirement-3

In mobile computing, the field's ability to analyze malware with good precession and recognition is necessary. There is a need to generalize malware within a specific malware family, such as the zero-day-attack family. It must be ensured that the malware detection system, when placed within mobile-based systems, should not consume too many computing resources.

3.4. Requirement-4

There is a need to consider all possible attacks and determine their counter-attacking system. The attacks through SMS messages can be classified based on the cause of failure, power consumption, information leakage, and financial charging. The attacking when it comes to financial charge is done through Vulnerability attack when mobile-based Micropayments are undertaken, payment rate service attack caused by the transmission of messages from behind without the user's notice, the transmission of large SMS messages, and through distributed denial-of-service (DdoS) attack. The attacks on mobile phones can be classified as hardware-based, vulnerabilities exposed by communication protocols, software-centric, which includes communication channels (short message service (SMS) and multimedia messaging service (MMS)), browsers, and the other category are the attacks that exploit through the user layer that include social engineering.

3.5. Requirement-5

Many malware detecting systems are in use which must be considered to enforce the same in the situations that need a specific detection method. Many techniques are in use for detecting. The authors have presented techniques to detect malware through static function call analysis, detections through signature, software-based attestation, and anomaly detection. They have also explained how mobile operating systems' protection could be achieved through isolation, hardened kernels, secure default settings, and software attestation for 3rd party apps.

3.6. Requirement-6

There is a need to consider the relationship between the artifacts of malware and non-malware programs. Many authors have proposed several approaches that focus on finding the relationship between malware programs related to artifacts and non-malware program-related artifacts and then establishing the relationship between the programs. Using the relationships, the programs that are affected due to malware can be reverse-engineered, and the malware removed instead of quarantining the affected program

3.7. Requirement-7

The dynamic behavior of operating systems, especially the Android operating system, needs to be considered. Many authors are cautiously working to detecting the existence of the malware while at the same, the attackers are inventing to find the anti-detection features making it complicated to find the existence of the malware.

3.8. Requirement-8

There is a need to collect information related to different malware in terms of its characteristics, features, and the pattern of occurrence, and store the same in a database. Hacker forums will help greatly in this regard to make available data/information about different malware. This database must be large enough that the example set will be large enough to properly predict a message to be malware if it is fitting into one of the formats or patterns of the messages stored in the database. It will be more advantageous to link a message stored in the database linked with the model used to classify the message.

3.9. Requirement-9

Machine learning (ML) techniques are being employed to understand the attacking approaches employed through malware. The ML techniques are being proved to be statistically sound for malware detection. Over time, the machine learning algorithms' ability to accurately detect malware's existence has been questioned due to the existence of too much vulnerability within the learning models that can be exploited by malware so that detection of the same is avoided.

3.10. Requirement-10

To take into account the techniques that cause code obfuscation. Anti-virus software is developed using signature detection techniques and is highly efficient when the malware and the signature that it creates are known quite ahead. The Antivirus software is not quite useful when the virus is not known quite ahead of its occurrence. Any new malware mainly focuses on changes, the sequence of execution of the code, registry renaming, replacing the instructions with equivalent instructions, reordering the instructions, and inserting code that collects the garbage collection.

The techniques used to make this kind of change are called code obfuscation. The kind of changes considered by each of the malware differs a lot and therefore is quite complicated to find the same. This kind of malware is called metamorphic malware, which focuses on making changes to the code, thus making the system work in unexpected terms.

3.11. Requirement-11

To exploit the communication features supported by the Android operating system so that handle on the incoming messages can be gained before the messages are transmitted to the respective applications. A broadcast receiver built into the operating system is a process that receives the messages and makes available the messages to the applications. Applications declare statically or dynamically their interest in receiving a certain type of information, and accordingly, the operating system (OS) will try to deliver the requested data when available.

3.12. Requirement-12

To exploit the buffers and logs maintained by the OS. Android has a special logging system through which the OS stores the logs in several circular buffers (for radio, events, and main). Developers can benefit from these buffers to get information from the system and debug their apps. For that purpose, a special command can be used in the ADB tool to extract data from the targeted buffer

3.13. Requirement-13

To study the behavior of existing malware systems. The malware application is designed to operate/behave like a regular application. It uses basic permissions required to send and receive short messages, process the messages, and extract sensitive information contained in those messages.

3.14. Requirement-14

The use of system resources must be continuously monitored. An app's behavior can be checked to find malware by monitoring the use of system resources, including power, memory, network, central processing unit (CPU), and based on information related to system calls, log information, and analysis of events are logged in Android-based systems. Frequent usage of the resource by the Applications holding on to the same for longer periods of time is clear indication of spurious processing taking place malware. Frequent Analysis of OS log laso will help indetetcing the malware that tend to overuse the resources.

3.15. Requirement-15

This is a need to continuously monitor the apps' behavior that is already installed and is running. Malware could enter into a system one way or another, making it necessary to continuously check if any malware entered into the system so that the malware app can be deleted. Frequesnt checking of the apps running under Android operating system is required to detetct the existence of the malware. There should be a system of querenting such malware after suspending the same from running.

3.16. Requirement-16

There is a need to implement the message dissecting system. The android system shall handover the received message to the App having the highest preference. The message is received by the highest preferred app using its communication interface, which is then dissected and classified into different applications. The app resposnibel for receiving and dissecting the message based on the type of application that initiated the message is registered with highest priority so that messgge is taken as input by the app that is responsible for dissecting the messages.

4. METHODS-ARCHITECTURAL DESIGN–MALWARE DETECTION, PREVENTION, AND MITIGATION

All the requirements mentioned above have to be modeled, and overall architecture containing the required software components must be developed. Figure 1 shows the top-level of malware identification, prevention, and mitigation systems. Two aspects of starting with must be modeled: continuous monitoring, code detection, and curing using different kinds of methods that include searching for a malware existence through either pattern-based detection or feature extraction and testing. Continuous monitoring is also required through analyzing the logs of the operating system. The users install many Apps from time to time, and any APP can initiate malware. In the future, if any of the monitoring methods are invented, the same must also be included in the monitoring model.

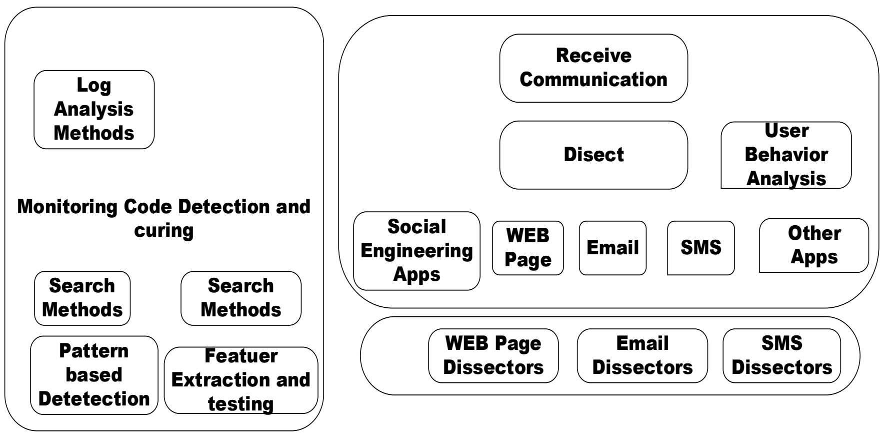


Figure 1. Top layer of comprehensive malware management architecture

The second aspect is to continuously inspect the incoming messages, carry the behavioral analysis, and classify the images into one category: social engineering app, web Surfing, emails, SMS, and other types of applications. Each class of the messages is further dissected based on the method to figure out the existence of the malware. This aspect of determining the method required is quite challenging. More than 12 methods have been proposed in the literature based on machine learning techniques. The speed of processing the incoming messages is the most important issue to be considered. The user will get stuck and get frustrated if the handheld device is mostly doing the backend processing to detect malware. The speed of processing is the most important criterion.

In the second layer of the architecture, the detectors of a particular type shown in layer-1 analyses the incoming message and learns the best model that must be used to detect the malware based on the message patterns, signatures, feature-based. The feature-based message detection is further analyzed to find the ML method that would be more suitable for detecting the malware occurrence. The models are learned based on example sets that are either universally or creating some examples based on the experts' experience. The models must be learned first before the same is used to detect SMS messages' existence. Figure 2 shows the components that act in the second layer of the comprehensive architecture. From Figure 2, it can be seen that SMS dissectors are required that classify the SMS messages into classes such as pattern-based detection, ML based detection and signature-based detection. The SMS message after having calssifed into a group, then will be analysed to find a suitable machine learnng method that is quite suitable for detecting the malware.

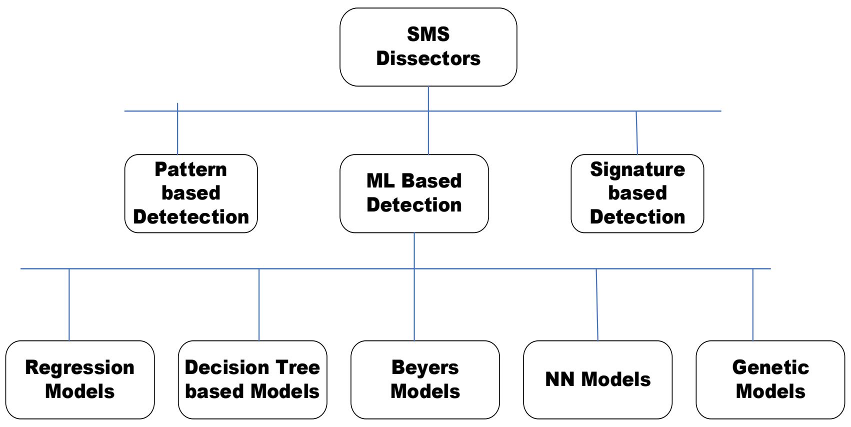
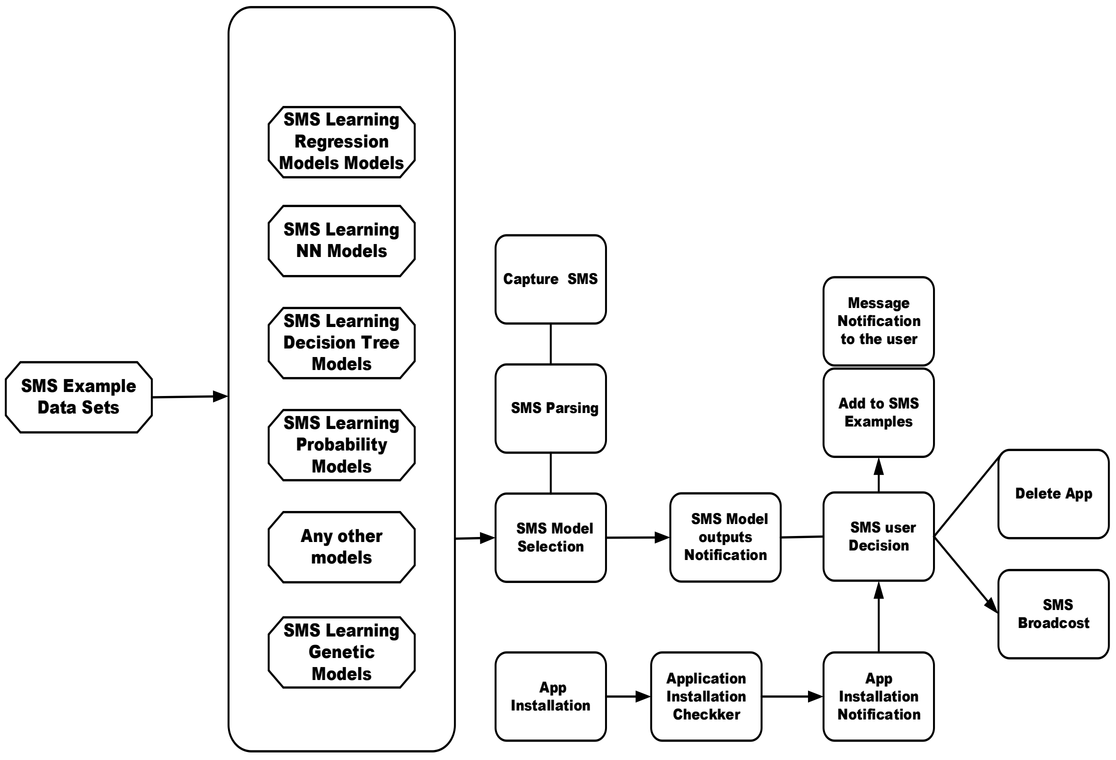


Figure 2. Second layer in the comprehensive architecture

The overall architecture that can be used for detecting, preventing amd mitigating the malware is shown in Figure 3. One segment of this architecture is related to learning different ML models using the example set. The learning models considered in the architecture include regression, numeral networks based, Bye's theory-based, decision three based, and genetic-based models. In the future, more learning models can be introduced.


SMS Example Data Sets
SMS Learning Regression Models Models
SMS Learning NN Models
SMS Learning Decision Tree Models
SMS Learning Probability Models
Any other models
SMS Learning Genetic Models
Capture SMS
SMS Parsing
SMS Model Selection
SMS Model outputs Notification
Message Notification to the user
Add to SMS Examples
SMS user Decision
Delete App
SMS Broadcost
App Installation
Application Installation Checkker
App Installation Notification
Figure 3. Overall comprehensive architectures for mitigating SMS based malware

Once the models are learned, the incoming SMS messages are parsed and analyzed to find the kind of learning model that is more suitable for detecting malware. The selected ML model then releases a notification indicting the malware's existence. Otherwise, the SMS decision module of the architecture adds the message to the example set if there is malware and sends out a notification to the user that there is malware in the message received. The decision module will broadcast the message if there is no malware in the SMS message. The Android operating systems will then handover the messages to the next app waiting in the queue as per the priorities.

The third segment of the architecture is detecting the malware while the App is installed. The Install checker will verify the app before installing the same; the decision-maker component deleted the app if notified by the install checker of the malware's existence in the app. Different methods that check the code is deployed to find the existence of the malware. Some of these methods look at the headers in the file structure of the binary programs.

5. RESULTS AND DISCUSSION

Several SMS malwares examples have been downloaded and database is created which are clearly mapped with the class to which the SMS message belongs, and the method used for detecting the malware. Some of the examples used are shown in the Table 1. Some test examples are taken and dseection of the same is done to recognize the class of the SMS message and the method that should be used to detect the malware as against the actual method used for detecting the malware. About 99% accuracy has been achieved. The Test examples that have been tested are shown in Table 2.

Table 1. Malware included SMS messages – example set

Serial Number	SMS Messagge	Class	Method is used
1	Hello this is my Bank Account number and the Audhar Card Number	ML Based Detection	Bayers Leraning Classifiers
2	Please click in the URL to know lattest information	ML Based Detection	Decision Tree Model
3	Down Laod the APP using the link and register to claim your gift	ML Based Detection	NN Model
4	Please install malware software or else your disk will be crashed	ML Based Detection	Genetic Algorithms
5	You have got a gift 12,00,000, claim the same	ML Based Detection	Regression models

Table 2. Disecting the SMS messages into class and method

Serial Number	SMS Messagge	Actual Class and Method		Predicated Class and Method	
		Class	Method is used	Class	Method is used
1	Please send Bank Account number and the Audhar Card Immediately	ML Based Detection	Bayers Leraning Classifiers	ML Based Detetcion	Bayers Leraning Classifiers
2	You have interesting information @ URL. Please Click.	ML Based Detection	Decision Tree Model	ML Based Detection	Decision Tree Model
3	You need to register at APP for gaining your Gift	ML Based Detection	NN Model	ML Based Detection	NN Model
4	Please register XYZ software for free and enjoy. Good videos and Images	ML Based Detection	Genetic Algorithms	ML Based Detection	Genetic Algorithms
5	You have got a gift 120,00,000, claim the same	ML Based Detection	Regression models	ML Based Detection	Regression models

6. CONCLUSIONS

Many roots exist for the malware occurrence with mobile systems that operate under the android operating system. One needs to plug all the holes to protect a Mobile based system from attacking. A comprehensive architecture is required to consider every root and mitigate the malware entering, deleting the malware that has been infringed into the system. In the architecture presented in this paper, malware entering through messaging initiated through different apps, malware detection at the time of installation of App, and eliminating the malware that infringed into the system through other means have been included. The method presented is comprehensive since every aspect of malware monitoring, prevention, detection, and elimination has been included in the architecture. The testing examples have been disected to map to the class and the method that should be used for detecting the malwre. 99% accuracy of dissection has been achived. Further research is in proress to learn the model and then move forowrd for predicting the existence of malware.

REFERENCES

[1] M. Belaoued, S. Mazouzi, S. Noureddine, and B. Salah, “Using Chi-Square test and heuristic search for detecting metamorphic malware,” *2015 First International Conference on New Technologies of Information and Communication (NTIC),* Mila, 2015, pp. 1-4, doi: 10.1109/NTIC.2015.7368758.

[2] D. Bruschi, L. Martignoni, and M. Monga, “Detecting Self-mutating Malware Using Control-Flow Graph Matching,” In: Büschkes R., Laskov P. (eds) Detection of Intrusions and Malware & Vulnerability Assessment. DIMVA 2006. Lecture Notes in Computer Science, vol 4064. Springer, Berlin, Heidelberg.

[3] S. Ramu, “Mobile Malware Evolution, Detection, and Defense,” *Semantic scholar*, 2012.

[4] H. S. Ham and M. J. Choi, “Analysis of Android malware detection performance using machine learning classifiers,” *2013 International Conference on ICT Convergence (ICTC)*, Jeju, 2013, pp. 490-495, doi: 10.1109/ICTC.2013.6675404.

[5] K. Hamandi, A. Chehab, I. H. Elhajj, and A. Kayssi, “Android SMS Malware: Vulnerability and Mitigation,” *2013 27th International Conference on Advanced Information Networking and Applications Workshops*, Barcelona, 2013, pp. 1004-1009, doi: 10.1109/WAINA.2013.134.

[6] Z. Aung and W. Zaw, “Permission-Based Android Malware Detection,” *International Journal of Scientific & Technology Research*, vol. 2, no. 3, pp. 228234, 2013.

[7] Y. Qu and K. Hughes, “Detecting metamorphic malware by using behavior-based aggregated signature,” *World Congress on Internet Security (WorldCIS-2013)*, 2013, doi: 10.1109/WorldCIS.2013.6751010.

[8] K. Blokhin, J. Saxe, and D. Mentis, “Malware Similarity Identification Using Call Graph-Based System Call Subsequence Features,” *2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2013, doi: 10.1109/ICDCSW.2013.55.

[9] K. Shaerpour, A. Dehghantanha, and R. Mahmod, “Trends in Android Malware Detection,” *Journal of Digital Forensics, Security and Law*, vol. 8, no. 3, pp. 21-40, 2013, doi: 10.15394/jdfsl.2013.1149.

[10] W. Park, K. Lee, K. Cho, and W. Ryu, “Analyzing and detecting Android malware method via disassembling and visualization,” *2014 International Conference on Information and Communication Technology Convergence (ICTC)*, Busan, 2014, pp. 817-818, doi: 10.1109/ICTC.2014.6983300.

[11] S. Yoon, J. Kim, and H. Cho, “Detection of SMS mobile malware,” *2014 International Conference on Electronics, Information and Communications (ICEIC)*, Kota Kinabalu, 2014, pp. 1-2, doi: 10.1109/ELINFOCOM.2014.6914392.

[12] K. Hamandi, A. Salman, I. H. Elhajj, A. Chehab, and A. Kayssi, “Messaging Attacks on Android: Vulnerabilities and Intrusion Detection,” *Journal of Mobile Information Systems*, vol. 2015, pp. 1-13, 2015, doi: 10.1155/2015/746930.

[13] Z. Wang, C. Li, Y. Guan, and Y. Xue, “DroidChain: A novel malware detection method for Android-based on behavior chain,” *2015 IEEE Conference on Communications and Network Security (CNS)*, Florence, 2015, pp. 727-728, doi: 10.1109/CNS.2015.7346906.

[14] S. Naval, V. Laxmi, M. Rajarajan, M. S. Gaur and M. Conti, “Employing Program Semantics for Malware Detection,” in *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2591-2604, Dec. 2015, doi: 10.1109/TIFS.2015.2469253.

[15] X. Zhang and Z. Jin, “A new semantics-based android malware detection,” *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, Chengdu, 2016, pp. 1412-1416, doi: 10.1109/CompComm.2016.7924936.

[16] S. Chen, M. Xue, Z. Tang, L. Xu, and H. Zhu, “StormDroid: A Streaminglized Machine Learning-Based System for Detecting Android Malware,” *ASIA CCS '16: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, May 2016, pp. 377-388, doi: 10.1145/2897845.2897860.

[17] A. Demontis, *et al*., “Yes, Machine Learning Can Be More Secure! A Case Study on Android Malware Detection,” *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 4, pp. 711-724, 1 July-Aug. 2019, doi: 10.1109/TDSC.2017.2700270.

[18] F. Di Cerbo, A. Girardello, F. Michahelles, and S. Voronkova, “Detection of Malicious Applications on Android O.S.,” *International Workshop on Computational Forensics*, 2011, pp. 138-149, doi: 10.1007/978-3-642-19376-7_12.

[19] J. Grisham, S. Samtani, M. Patton and H. Chen, "Identifying mobile malware and key threat actors in online hacker forums for proactive cyber threat intelligence," *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Beijing, 2017, pp. 13-18, doi: 10.1109/ISI.2017.8004867.

[20] B. Amro, “Malware Detection Techniques for Mobile Devices,” *International Journal of Mobile Network Communications & Telematics (IJMNCT)*, vol. 7, no. 46, December 2017, doi: 10.5121/ijmnct.2017.7601.

[21] M. Yousefi-Azar, Leonard G. C. Hamey, V. Varadharajan, and S. Chen, “Malytics: A Malware Detection Scheme,” *IEEE Access*, vol. 6, pp. 49418-49431, 2018, doi: 10.1109/ACCESS.2018.2864871.

[22] M. N. Arif, A. A. Bakar, and M. M. Saudi, “A New Mobile Malware Classification for Audio Exploitation,” *International journal of engineering and technology*, vol. 7, no. 4, pp. 59-62, 2018, doi: 10.14419/ijet.v7i4.15.21372.

[23] A. A. D. Utku, and M. A. Akcayol, “Decision tree-based android malware detection system,” *2018 26th Signal Processing and Communications Applications Conference (SIU)*, Izmir, 2018, pp. 1-4, doi: 10.1109/SIU.2018.8404151.

[24] S. Iqbal and M. Zulkernine, “SpyDroid: A Framework for Employing Multiple Real-Time Malware Detectors on Android,” *2018 13th International Conference on Malicious and Unwanted Software (MALWARE)*, Nantucket, MA, USA, 2018, pp. 1-8, doi: 10.1109/MALWARE.2018.8659365.

[25] S. Jammalamadaka and T. B. Mirilaya, “Securing Multi-tenancy systems through multi D.B. instances and multiple databases on different physical servers,” *International Journal of Electrical and Computer Engineering*, vol. 9, no. 2, pp. 1385-1392, 2019, https://doi.org/10.11591/ijece.v9i2.pp1385-1392

[26] M. Trinath Basu and JKR Sastry, "Improving the OpenStack Authentication system through federation wit h JASON Tokens," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 6, pp. 3596-3614, 2019, doi: 10.30534/ijatcse/2019/143862019.

[27] M. Trinath Basu and JKR Sastry, "Strengthening Authentication within OpenStack Cloud Computing, System through Federation with ADDS System," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 1, pp. 213-238, 2020, doi: 10.30534/ijeter/2020/29812020.

[28] JKR Sastry and M. Trinath Basu, "Multi-Factor Authentication through Integration with IMS System," *International Journal of Emerging Trends in Engineering Research*," vol. 8, no. 1, pp. 88-113, 2020.

[29] JKR Sastry and M. Trinath Basu, "Securing SAAS service under cloud computing-based multi-tenancy systems," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 1, pp. 65-71, 2019, doi: 10.11591/ijeecs. v13.i1.pp65-71.

[30] M. T. Basu and JKR Sastry, "Enhancing Data Security under Multi-Tenancy within OpenStack," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 1, pp. 533-544, 2020, doi: 10.30534/ijatcse/2020/73912020.

[31] JKR Sastry and M. Trinath Basu, "Enhancement of Security within OpenStack – Some measures," *International Journal of Emerging Trends and Engineering Research*, vol. 8, no. 3, pp. 919-938, 2020, doi: 10.30534/ijeter/2020/49832020.

[32] JKR Sastry and B. Trinath Basu, "Implementing user defined Attribute and Policy based Access Control," *International Journal of Emerging trends in Engineering Research*, vol. 8, no. 7, July 2020, pp. 10.30534/ijeter/2020/171872020.

[33] B. Panda and S. N. Tripathy, "Morphological malware detection: An API sequence mining with LCS based voting algorithm," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 10, no. 6, pp. 625-623, 2018.

[34] V. M. Venkateswara Rao and A. Anand Kumar, "Artificial Neural Network and Adaptive Neuro Fuzzy Control of Direct Torque Control of Induction Motor for Speed and Torque Ripple Control," *Proceedings of the 2nd International Conference on Trends in Electronics and Informatics*, ICOEI 2018, pp. 1416-1422, doi: 10.1109/ICOEI.2018.8553871.

[35] R. Patil and C. Prakash V., "Neural network-based approach for improving combinatorial coverage in combinatorial testing approach," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 20, pp. 6677-6687, 2018.

[36] M. Arshad and A. Hussain, "A Novel multi-level attack detection and prevention model for Dynamic LAN/WLAN networks," *Revista Tecnica de la Facultad de Ingenieria Universidad del Zulia*, vol 41, no. 1, pp. 59-66, 2018.

[37] S. Cheerla, D. Venkata Ratnam, K. S. Teja Sri, P. S. Sahithi, and G. Sowdamini, "Neural network based indoor localization using Wi-Fi received signal strength," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 10, no. 4, pp. 374-378, 2018.

[38] P. V. Vara Prasad, N. Sowmya, K. Rajasekhar Reddy, and P. Jayant Bala, "Introduction to dynamic malware analysis for cyber intelligence and forensics," *International Journal of Mechanical Engineering and Technology*, vol. 9, no 1, pp. 10-21, 2018.