ISSN 2347 - 3983
Volume 8. No. 10, October 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter038102020.pdf
https://doi.org/10.30534/ijeter/2020/038102020

A Comprehensive architecture for protecting Android Based Applications from Malware affected through SMS messages

Dr. Sastry JKR[1], T. Chandrasekhara Reddy[2]
[1]KLEF Deemed to be University, Vaddeswaram, India, drsastry@kluniversity.in
[2]KLEF Deemed to be University, Vaddeswaram, India, chandu.tummuru@gmail.com

ABSTRACT

Android based messaging systems are being used heavily these days by the people around the world for effecting the communication. Android based applications communicate with the users for emailing, text messaging, and transmission of Audio and Videos objects. The actual email, text, videos or Videos are manipulated either at the transmitting stage or at the intermediate stage especially at the back end of processing such that a runtime routine is invokes which cause the damage to the Local resources of the Mobile Phone.Many issues are involved in dealing with the malware detecting, prevention and curing if the malware entered into the system from different emails

In this paper, a comprehensive architecture is presented that considers every issue relating to the Malware especially in relation to malware affected through SMS messages operated under Android operating System.

Key words: Android, Malware, Messaging systems, Malware detection, Comprehensive Architecture

1. INTRODUCTION

Smart phones are being used for many purposes that include playing Games, online information access, dealing with emails and messages, surfing the net, making commercial transactions. The transactions carried by the users involve personal information that is stored in address book and within SMS messages which has lead to opening up many security concerns. Many smart phones are being released from day to day and therefore have become target for the attackers for distributing malware and perform malicious attacks.

Malware is a malicious code that can do anything in any other program can such as writing a message, stopping a running program, modifying a file etc. Also, malware can be triggered periodically or lie dormant undetected until some event triggers the code to act. They are further classified as Trojans, bots, virus, backdoor, worms, rootkits etc.

Malware is software which is developed to write messages managing the operations of the programs, ability to modify the files. Malware can be made to inactive for some time and can be made to be running at periodic times or invoked when certain events happens within the smart phone systems.

Malware exists in variety ways that include Virus, worms, Trojans, bots, rootkits, backdoor etc.

Android is behind every device that makes just working to actually make life easier. Android is the reason your GPS avoids traffic, your watch can text and your Assistant answers questions. It's the operating system inside 2.5 billion active devices. 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. Sometime 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 breach existing exposed by android operating system

Mobile applications are the most widely used by many of the users in the world. The usage of smartphones and mobile applications is increasing more and more. Presently smartphone applications are used frequently by the users. Security is most critical issue when it comes smartphones because most of the smart phones are developed using open architecture. Many of the users world overuse their 80% of time doing messaging using smart phones. Attackers manipulate the message by taking control of the message while they are in transmission to the Messaging server situated at a far of location,

The Attackers generally inserts URL links within the messages and the users by chance clicks on the link which actually leads to down loading the small programs and the same is installed and executed. This kind of programs are intelligent that can find the complete the operating environment and then finds the critical operations that when initiated damages major portion of 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 to a system with a USB or can 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 the

use of a command and control server that allows attackers to communicate with the infected systems, infiltrate data and also remotely control the compromised device. The practice in which an attacker sends emails purporting to be from reputable companies in order 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 with the goal of diverting the user to a web site which collects the data about the users and creates a record having the data collected from the user. The data is then used to for checking and deriving the most confidential information through brute force methods. Sometimes programs are downloaded as popup programs and the same are executed as background processes when allowed by the users. Users always are hard-pressed asking the users to enable the POP ups so that the page that they are looking is promptly displayed.

Sometimes the users are prompted to subscribe for the site for better use of the facility and collect important information related to the users in the name of the registrations and uses the same for attacking. After registration is done, the users are displayed with wen sites that is provided with the contact information which can be used by the user either to send an email or make call. The attacker collects details in the name of authentication and then start using the same to login to different sites especially to the sites that are related to financial and e-commerce transactions. The malicious web sites lure the users in the name of discounts promotional code, subscriptions, coupons for getting discounting etc.

There are many such ways of attacking the mobile applications as annunciated in the literature. There is continuous requirement of monitoring the behavior of ongoing Applications and find is any malware is operating and mitigate the same if exists. There all also continuous requirement 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 the malware entering through different processes. There are many working situations that lead to malware and a different method needs to be considered for detecting and mitigating the malware. In this paper a comprehensive architecture that helps mitigating malware considering every source and process through which it enters. The architecture is designed to consider detection, prevention and eliminate the malware form mobiles which are operated through android operating system.

2. RELATED WORK

.

Anti-Virus Software are developed using signature detection techniques and have been found to be highly efficient when the malware and the signature that it creates are know quite ahead. The Antivirus software is not quite useful when the kind of the Virus is not known quite ahead of it occurrence.

Any new Malware mainly focusses on changes the sequence of execution of the code, registry renaming, replacing the instructions with equivalent instructions, reordering the instructions and insertion of code that collects the garbage collection. The techniques used to make this kind of changes are called as code obfuscation. The shape of changes considered by each of the malware thus differs a lot and therefore is quite complicated to find the same. This kind of malware is called metamorphic malware which focus on making changes the code thus making the system to work in unexpected terms.

M. Belaoued et al [1] have proposed a method for detecting the malware which is based on detecting the program file headers as there will be change in the headers when code changes takes place. They have used a hybrid filter wrapper to extract the characteristics / features that are most relevant and match the characteristics of the data. They have used Chi-Square test for removing the irrelevant data Out of the filtered characteristics; they have filtered most relevant features using Greedy Hill Climbing search Method. The data subsets generated by the Greedy Hill climbing search are then subjected to learning algorithms that classify the data set into one of the malware variants. They have experimented the technique using windows PE files which are situated as headers in each of the program.

The polymorphic and metamorphic techniques are mostly based on pattern matching, as each variant is focussed on making changes to the code using a different pattern as presented by Bruschi D et al [2]. Pattern matching is regularly used to find if there is any threat. The whole of the issue is to check the existence of the malware in the code from time to time. The effected programs are found and quarantined. Malware detectors of different types to find the existence of different variant of malware. Bruschi D et al., have proposed a method that controls the program flows of a programs with the control flows of the programs which are affected with malware.

Ramu et al., [3] have presented a survey on the way Mobile Malware have been evolved , and the way the malware is detected and the way the mobile system is protected when attacking is done using malware. They have narrated different malware that have been introduced into mobile phones that operate on android based mobile systems. Availability of huge documentation that explains the way Android OS operate has led to introduction of malware many ways.The huge popularity of Android, freely available documentation on Android platform and weak screening process of Android marketplace were attributed to this surge in malware attacks. The attacks on Mobile phones can be classified as hardware based, vulnerabilities exposed by communication protocols, Software centric which includes Communication channels (SMS and MMS), Browsers and

the other category are the attacks that exploits through the user layer that include social engineering. The authors have presented techniques to detect the existence malware through static function call analysis, detections through signature, software based attestation, anomaly detection etc. The have also explained the way the protection of mobile operating system could be achieved through process of isolation, hardened kernels, secure default settings, software attestation for 3rd party apps.

Understanding the malware characteristics and Android Architecture is essential and focusing on unnecessary features just to leads to wastage of time and resources. Machine learning classifiers have been used to detect the existence of the malware. The input to the classifiers is fed through many resources that include system calls, log information, analysis of events which are logged in android based systems. Hyo-Sik Ham et al [4] have proposed a new feature set and used the same to detect the malware by inputting the feature set to machine learning classifiers. They have vectored the use of various kind of resources such as Power, memory, network, CPU by various apps and used a ML classifier to detect existence of the malware.

Many messaging design vulnerabilities exist in the Android Operating System. Most of the malware applications exploit theses vulnerabilities. The malware applications is designed to operate / behave like a regular application and uses basic permissions required to send and receive short messages, process the messages and extract sensitive information contained in those messages. The android supported Broadcast receiver, messaging sending, and the permission system supported by android operating system are used by the Malware Messaging system and gains absolute control over the SMS messages for transmitting, receiving and hiding the SMS messages. The malware application also hides any acknowledgements transmitted by the telecom operator. The malware application thus can cause series of malicious transactions affecting the users who operate using the messages. K. Hamandi et al., [5] have demonstrated working of such an application.

Many Android based application can be developed and installed on a Mobile phone. There is no third party verification and validation of such APPS. Thus the APPS installed on the Android based Mobile phone may contain malware. The malware applications can be written to steel most important information of the users and thus can attack.Zarni Aung et al., [6] have proposed a framework to detect the existence of malware within the APPS. The framework is based on machine learning based malware detection. The system is built based on permissions provided to the apps and the kind of events generated by those applications.

Metamorphic and polymorphic malware are making it complicated to detect the malware. The signatures and pattern of signatures keep varying dynamically making complicated to detect the malware. Most of the signature based detection methods have failed as only signatures based on executable code files which are obfuscated are investigated. The signature patterns keep changing from time to time. However the behaviour of the advanced malware is consistent throughout many variants of the signatures. Finding signature that is cons detent across a family of malware helps in quickly detecting the malware. Yanzhen Qu et al [7] have presented that aggregating many signatures and using the same to detect the malware will be quite helpful especially in terms of reducing the signature database.

Many authors have proposed several approaches that primarily focus on finding the relationship between a malware programs related artifacts and a non-malware program related artifacts and then establish 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 effected program

Kristina Blokhin et al [8] have presented a code sharing analysis method that focus on partitioning the malware call logs into system call subsequence, through identification of the locations in the logs where the set of saved instruction pointers on the program call stack changes significantly. Theses sub-sequences are the sub-sequence of the system calls that can be represented as the local regions of a program call graph. A computing similarly matrix thus can be developed using the sub-sequences as the features. Similarity analysis is carried to find the program under question is a malware or not

KavehShaerpour et al [9]have surveyed different techniques prosed in the literature for detecting the malware. The techniques surveyed by them include malware detection using host based frameworks, static analysis of executable, feature analysis and behavioural patterns. Comparison of all the techniques has been carried considering the features used in each of the technique. They have also discussed the kind of challenges that have been faced in the development and implementation of the techniques. They have also discussed methods that can be used for detecting malware under Android operating system.

The android based applications have been in rapid increase. There has been a ramped increase in introducing different kinds of malware. W. Park et al. [10] have presented method that focus on finding the similarity of the Applications with the existing malwares that run Android platform. They have focused on finding the visual similarity among android based malware and they have also focused on finding the degree of similarly among the malware families. Once the similarities are known, one can decide on the number of detectors required to detect whether a malware is coming in.

S. Yoon et al. [11] have analyzed financial transactions that are caused using SMS and the users start attacking using the data contained in the SMS and have proposed a method that monitors the malicious user behavior and have described malicious analysis techniques to detect the attacks. They have presented the methods related to malware installation checking, Analysis of Messages which are either transmitted or received, signature based pattern matching.

The attacking through SMS messages can be classified based on 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 Micro payments are undertaken, payment rate service attack caused through transmission of messages from behind without the notice of the user, Transmission of large SMS messages, and through DDoS attack.

Khodor Hamandi et al. [12] have presented different kinds of attacks on the messaging framework supported by Android. They have focused on SMS and USSD and the kind of security provision made into the Android Operating System. They have described different android elements that are responsible for different types of attacks. They have also presented architecture of intrusion detection system that thwarts SMS messaging attacks.

They have presented different elements that cause security issues when SMS based applications are to be built. The components include GUI which facilitates the GUI, components that run as background processes which can start a service, providing permissions for the applications that provide access to android store, notifying the user and guarding the critical APIs.

Broadcast receiver 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 OS will try to deliver the requested data when available.

For the procedure of sending information, Android uses "Intents" which are data structures that should be passed to "sendBroadcast", for example. Android defines two types of Broadcast Receivers: normal and ordered. The normal ones are asynchronous so there is no defined order according to which apps would receive the data. As for the ordered ones, a priority can be set to require from the system to deliver the information to each app in a certain sequence, and as such, some apps will get the information before others. This feature allows developers to capture and possibly modify the carried data before it reaches lower-priority consumers. In this case, an app can prevent other apps from getting specific data by aborting the received data.

According to our experiments, for an app to ensure that it will obtain an ordered broadcast, it has to be the first application to register to the desired intent with the highest priority. It is worth noting that an app can register for intent with any priority it specifies with no constraints or limitations after being given permission.

The SMS manager, part of the Android telephony stack, provides developers with the necessary functions to send messages. In order to send a text message, and apart from getting the right permissions, an app can send a text message at any time by a simple function call. The main function to send SMS messages is "sendTextMessage". Calling the send function displays no notifications on the phone; the sending process is seamless and transparent to mobile users.

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 (*logcat*) can be used in the ADB tool to extract data from the targeted buffer

Zhaoguo Wang et al. [13] have proposed a method called DroidChain to fight against malware variants and zero day malware. The model is based on behavioral chains of APPs running under Android. They have summarized four kind of detection models that include Privacy Leakage, SMS financial charge, Malware Installation and privilege escalation.

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 complicated to find the existence of the malware. It has been proved that the models which try to detect the existence of the Malware using dynamic behavior models outperform signature based detecting methods by neutralizing the effects due to code obfuscation or morphing.

The dynamic behavior based models are based on system calls to model the propagation and infection dynamics of the malware. But these approaches do not consider the anti-detection feature of the Modern malware which causes at another attack called anti System call injection attack. This kind of attack allows the malicious binaries to inject independent and irrelevant system calls during program execution leading to changing the execution sequence making the existing system call based detection ineffective.

To counter the System-call injection method S. Naval et al. [14] have proposed a method that characterizes program semantics using Asymptotic Equipartition Property (AEP) which is generally applied to information theoretic domain. The approach quantifies the call sequences based on the extent to which the sequences can detect malicious binaries. This technique is not vulnerable to call injection attacks as these components are not visible to malware authors. Furthermore, the proposed detection model is less vulnerable to call-injection attacks as the discriminating components are not directly visible to malware authors.

The modern malware are designed to employ anti-detection techniques such as code obfuscation. The apps that are built to take care of incoming malware are vulnerable to common code transformation techniques.Xiaohan Zhang et al. [15] proposed an enhance malware detection method that combines both static analysis and use of ensemble techniques for improving the malware detection accuracy. The model proposed by them extracts semantic based features which can resist code obfuscating techniques and also features from code and app characteristics through static analysis.

Sen Chen et al. [16] have proposed a framework called StroDroid which is built with streaminglized machine learning caused through enhanced features observed through a large collection of data set. They have streaminglized the whole MD process to support large-scale analysis, yielding an efficient and scalable MD technique that observes app behaviors statically and dynamically.

Machine Learning 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 the time the ability of the machine learning algorithms to accurately detect the existence of malware has been questioned due to existence of too much vulnerability within the learning modelsthat can be exploited by malware so that detection of the same is avoided. A framework exists that help categorising the ML based malware detection considering the potential attacking scenarios so that ML based detection systems are built with varying features and capabilities.

A. Demontis et al. [17] propsed "Drebin" a malware detector that implement 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.

Di Cerbo F et al. [18] have presented a methodology for carrying forensics analysis for detecting malicious malware based applications. The methodology is built on top of existing security enforcement features used within Android Operating system especially considering the permissions exposed by those applications.

Most effort is being spent in dealing with internal protection through use of antivirus software which finds whether any file or program existing in computing system is infected with the Virus. This is like a postmortem work, as the virus if has already entered would have damaged some part of the system before an attempt is made to remove the same. However any incoming program / data must be verified to check the existence of the Virus before same are allowed to get into computing resources. To achieve this there is a need to collect information about different malware in terms of its characteristics features, pattern of occurrence etc. and store the same in a database, Hacker forums will help greatly in this regard to make available data / information about different malware.

Malware is easily spread when the users interact with Social engineering sites through making available attachments of different types to be stored within the sites. J. Grisham et al. [19] have used recurrent neural networks for identifying the infected attachments and then carrying the social network analysis for finding the key hackers disseminating from mobile devices. It has been found from this study that many of the zipped attachments are actually Apps that are actually the malware made by the hackers

Most of the information processing is shifted to Mobile phones these days due to portability and high performance. Rapid hardware evolution is taking place for the development of different kinds of mobile phones having different kinds of interfaces. Both software and hardware design focused on increasing performance and the working hours of a mobile device. There are primarily two operating systems that evolved to drive the mobile phone which includes iOS and Android

Mobile phones also are being attacked by the mollified attackers. There is need to provide protection within each and every mobile phone so that no software that damage the system could creep inside the mobile phone. Belal Amro et al. [20] have analyzed different malware detection techniques that suit mobile operating systems like Android and iOS. They have also provided an easement of each of the techniques. The have carried on this work to provide a platform to detect malware based on user profiling.

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

MAHMOOD YOUSEFI-AZAR et al. [21] have presented a method "Malytics" that focus on extracting static features of any binary file to distinguish malware. The method is developed using three stages which include feature extraction, similarity measurement, and classification.All the three phases are implemented within a neural network having two hidden layers and one output layer. The feature extraction performed through tf-simhashing is undertaken in the first layer of a neural network

Malware enter into Mobile phones via Audio, SMS, call logs, images and videos. Malware enters into a mobile differently suing different media. Muhamad Nur 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.

Muhamad Nur Arif et al. [22] have adopted a method that help analysing the audios and determine 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 A 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 Android operating system.

As the number Android based Mobile apps are increasing the number of malware that effects the Apps are also increasing. Many techniques have been invented to detect the malware, boy few them deal with real time monitoring of the devices of the devices as Android does allow low level information to any of the APPS introduced by third party service providers. Some techniques detect the malware much better than the other. There is a need to detect employ many detectors so that any kind of malware coming in can be detected.

S. Iqbal et al.[25] have presented a real time monitoring and detection technique called SpyDroid which employs multiple detectors developed by third parties for detecting the malware and also allows for continuous monitoring of the devices in real time.

They have introduced two modules within an APP for monitoring and detection. The monitoring module provides an interface to other third party Malware detecting APPs that monitor and analyses runtime information of all the running APPS and provides information to the detecting module about the existence of malware or otherwise to the detecting Module. The detection module then decides the time at which a particular Application based APP is to be marked as Malware. End users can install any number of APP detectors. This way, more number of detectors can be introduced as they are invented and the detecting module can take a decision on other Application APPs for declaring as Malware.

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[26][27[28][29][30][31][32][33][34][35][36][37][38][39]. Panda et al. [40] have presented morphological malware detection method where the signatures of malware keep changing from time to time. Many Neural networks based learning methods have been presented which are connected with malware detection and prevention [41][42][43][44][45][46][47][48].

3. GAP ANALYSIS

Many Malware detection methods exist that actually 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 works perfectly in some contexts. As the contexts changes, the methods become in effective. Very few frameworks are available that consider all possible methods that detect Malware in a given context

Android security system does not allow direct access to the device resources but provides system call interface that can be used to get the status of each of the APP especially the system resources used by those APPS.

If checking for SMS is done while the messages are communicated, there would be heavy delay which hampers functioning of many other APPS that depend on the messages received by the Messaging APPs. While Malware through SMS messages can be detected through malware detectors, malware can still be coming in through other mechanisms such as email.

Thus there should be a continuous monitoring system that detect the existing malware and then quarantine the apps that are found to be malware while at the same keep checking instream mechanisms that can produce malware through SMS messages and other means.

Thus a mechanism is required that keep monitoring continuously the existence of the malware and there should be a method that keep detecting the incoming messages to find whether it contains any malware in it. Probability based models are more realistic when it comes to continuous information coming through SMS messages.

In this paper architecture is presented that provides overall structure to deal with instream Malware detection that can enter through Emails, SMS messages, Surfed Messages and Communicating data structures and also continuously monitoring the existence of Malware effected are malware software functioning within the Mobile Phones.

This paper presents the Extensive architecture that deal with handing Malware that can arise through any of the sources.

4. COMPREHENSIVE ARCHITECTURE FOR DEALING WITH MALWARE

The malware applications is designed to operate / behave like a regular application and uses basic permissions required to send and receive short messages, process the messages and extract sensitive information contained in those messages.

The android supported Broadcast receiver, messaging sending, and the permission system supported by android operating system are used by the Malware Messaging system and gains absolute control over the SMS messages for transmitting, receiving and hiding the SMS messages. The malware application also hides any acknowledgements transmitted by the telecom operator. The malware application thus can cause series of malicious transactions affecting the users who operate using the messages.

For the procedure of sending information, Android uses "Intents" which are data structures that should be passed to "sendBroadcast", for example. Android defines two types of Broadcast Receivers: normal and ordered. The normal ones are asynchronous so there is no defined order according to which apps would receive the data. As for the ordered ones, a priority can be set to require from the system to deliver the information to each app in a certain sequence, and as such, some apps will get the information before others. This feature allows developers to capture and possibly modify the carried data before it reaches lower-priority consumers. In this case, an app can prevent other apps from getting specific data by aborting the received data.

The behaviour of an App can be checked to find if it is malware by monitoring the use of system resources which includes power, memory, network, CPU, and based on information related to system calls, log information, analysis of events which are logged in android based systems. This data can be made to a learning model which can predict whether the App using the system resources is a malware or otherwise.

There are two aspects that needs to be looked into which include continuous monitoring for finding and eliminating the resident malware which have been infringed into the system in one way or other.

Malware could enter into a system one way or other, making it necessary to continuously check if there any malware entered into the system so that the malware app can be deleted. Many methods have been presented in the literature as on today and many more methods shall be introduced in future for disecting the existence of malware. The methods presented are bases on pattern recognition, feature recognition, searching for the existence of hey features etc. When method is to make all the methods find the malware and the Apps that are voted by many detection methods can be declared to be malware and therefore stands to be elimanted.

The second aspect is detecting malware if any entering through the messages that flow across various Apps and the Users. User communicate with each other through transmission of web pages, emails, SMS messages, the Apps used by social engineering apps that are used to exchange the messages, videos and Audios. The messages that contain text, audios and videos will be transmitted in different format and structures. There is need to dissect each of the message and find its source. The App must have highest preference for the message which is passible by setting highest preference say 999 so that the android system service that is responsible for transmitting and receiving through its system calls broadcast ( ) and recvmsg ( ) will handover or receive the messages from any of the App that has the highest preference. 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 then is dissected and classified into different types of applications that include Social engineering App, a web surfing App, Apps that deal with emails and SMS meshes and other applications. The classified messages then are analyzed and classified using any of the learning models implemented by different kinds of dissectors which include web page directors, email dissectors and SMS dissectors. The dissectors classify the incoming message into different categories based on the type of method used for detecting the existence of the malware. If malware is existing users are informed accordingly and if no malware is existing, the App broadcast the message back by calling broadcast( ) system call in which case the android operating system and hands over the message to the App that has next preference. Figure-1 shows the Architecture in the first layer of the comprehensive architecture

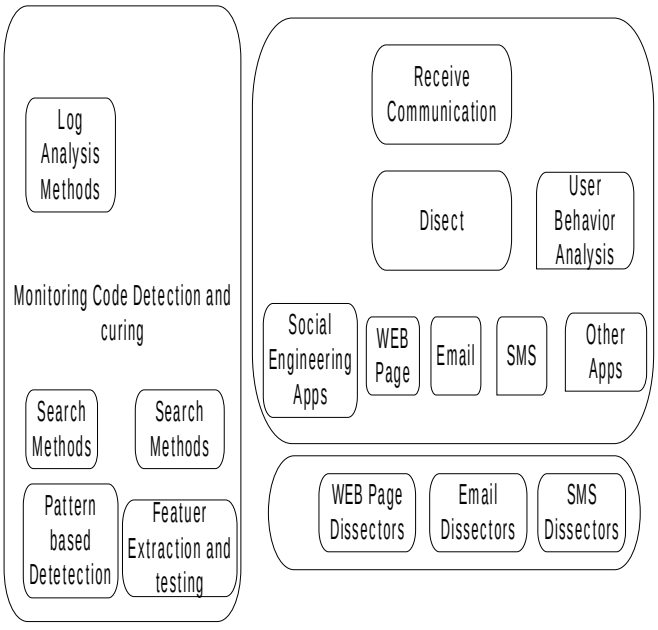

Figure 1: Top layer of comprehensive Malware Management architecture

In the second layer of the architecture, the detectors of 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 occurrence of the Malware. The models are leant based on example sets that are either universally or creating some examples based on experience of the experts. The models must be learnt first before the same are used to detect the existence in the SMS messages. **Figure 2** shows the components that act in the second layer of the comprehensive architecture.

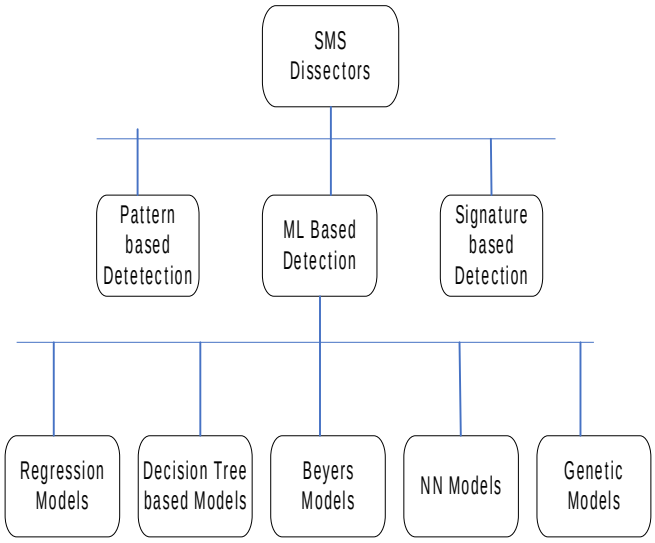

Figure 2 : Second Layer in the Comprehensive Architecture

The overall architecture is shown in **Figure 3**. One segment of this architecture is related to learning different kind of ML models using the example set. The learning models considered in the architecture include regression, Numeral networks based, Byes theory based, decision three based and genetic based models. In future more learning models can be introduced.

Once the models learnt, the incoming SMS messages are parsed and analyzed to find the kind of learning model that is more suitable for detecting the malware. The selected ML model then releases a notification indicting the existence of the malware or otherwise The SMS decision module of the architecture adds the message to the example set if there is a malware and sends out a notification to the user that there is a malware in the message receive. The decision module will broad cast the message if there is no malware in the SMS message. The android operating systems will then handover the messages to next App waiting in the Queue as per the priorities.

The third segment of the architecture is the 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 the existence of the malware in the App. Different methods that checks the code is deployed to find the existence of the malware. Some of these methods look at the headers that are contained in the file stricture of the binary programs.

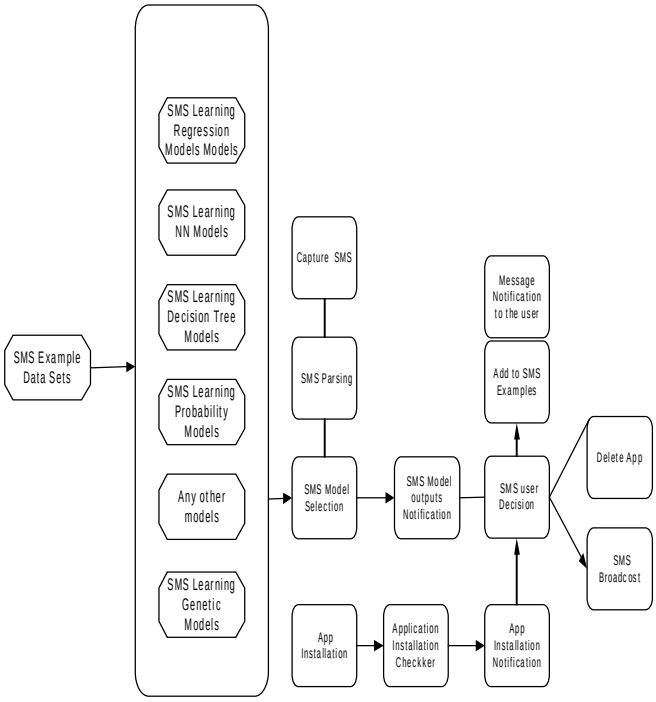

Figure 3: Overall comprehensive architectures for mitigating SMS based malware

5. CONCLUSIONS

Many roots exist for the occurrence of the malware with mobile systems that operate under android operating system. One needs to plugs all the holes for protecting a Mobile based systems from attacking. A comprehensive architecture is required that considers every root and mitigates 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 and detection and eliminations has been included into the architecture.

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] Bruschi D., Martignoni L., Monga M. (2006) 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. https://doi.org/10.1007/11790754_8

[3] Ramu, S.. "Mobile Malware Evolution, Detection and Defense." (2012), Sematic scholar.

[4] Hyo-Sik Ham and Mi-Jung 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] Zarni Aung, Win Zaw, Permission-Based Android Malware Detection, INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH, VOLUME 2, ISSUE 3, MARCH 2013

[7] Yanzhen Qu, Kelly Hughes, Detecting Metamorphic Malware by Using Behavior-based Aggregated World Congress on Internet Security (WorldCIS-2013)

[8] Kristina Blokhin, Josh Saxe, David Mentis, Malware Similarity Identification Using Call Graph Based System Call Subsequence Features, 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops

[9] KavehShaerpour, Ali Dehghantanha, RamlanMahmod, TRENDS IN ANDROID MALWARE DETECTION, Journal of Digital Forensics, Security and Law, Vol. 8(3)

[10] W. Park, K. Lee, K. Cho and W. Ryu, "Analyzing and detecting method of Android malware 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] KhodorHamandi, Alaa Salman, Imad H. Elhajj,Ali Chehab,[1] and Ayman Kayssi, Messaging Attacks on Android: Vulnerabilities and Intrusion Detection, Journal of Mobile Information Systems, Volume 2015, https://doi.org/10.1155/2015/746930

[13] Zhaoguo Wang, Chenglong Li, Yi Guan and YiboXue, "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] Xiaohan Zhang and ZhengpingJin, "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] Sen Chen, MinhuiXue, Zhushou Tang, Lihua Xu, Haojin Zhu, Authors Info &AffiliationsASIA CCS '16: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, May 2016 Pages 377–388https://doi.org/10.1145/2897845.2897860

[17] A. Demontis et al., "Yes, Machine Learning Can Be More Secure! A Case Study on Android Malware Detection," in 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] Di Cerbo F., Girardello A., Michahelles F., Voronkova S. (2011) Detection of Malicious Applications on Android OS. In: Sako H., Franke K.Y., Saitoh S. (eds) Computational Forensics. IWCF 2010. Lecture Notes in Computer Science, vol 6540. Springer, Berlin, Heidelberg. https://doi.org/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] BelalAmro, Malware detection techniques for mobile devices, International Journal of Mobile Network Communications & Telematics (IJMNCT) Vol.7, No.4/5/6, December 2017

[21] Mahmood Yousefi-Azar, Leonard G. C. Hamey, Vijay Varadharajan, Shiping Chen, Malytics: A Malware Detection Scheme, IEEE Access, 2018

[22] Muhamad NurArif and Azreena Abu Bakar and M. M. SaA New Mobile Malware Classification for Audio Exploitation}, International journal of engineering and technology, 2018, Volume 7 Pages 59-62

[23] Utku, A. A. Doğru 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] JKRSastry, M TrinathBasu, Securing Multi-tenancy systems through user spaces defined within the database level, Jour of Adv Research in Dynamical & Control Systems, Volume 10, issue 7, Page 405-412, 2018

[26] JKRSastry, M TrinathBasu, Securing Multi-tenancy systems through multi DB instances and multiple databases on different physical servers, International Journal of Electrical and Computer Engineering e 2, Pages 1385-1392, 2019, https://doi.org/10.11591/ijece.v9i2.pp1385-1392

[27] M.TrinathBasu, Dr.JKRSastry, A full security included Cloud Computing Architecture, International Journal of Engineering & Technology, Volume 7, Issue 2.7, Page 807-812, 2018

[28] M. TrinathBasu, JKRSastry, Improving the OpenStack Authentication system through federation with JASON Tokens, International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, Issue 6, Pages 3596-3614,2019 https://doi.org/10.30534/ijatcse/2019/143862019

[29] TrinathBasu, JKRSastry, Strengthening Authentication within OpenStack Cloud Computing, System through Federation with ADDS System, International Journal of Emerging Trends in Engineering Research, Volume 8, No. 1, Page, 213-238, 2020 https://doi.org/10.30534/ijeter/2020/29812020

[30] JKRSastry, M TrinathBasu, Multi-Factor Authentication through Integration with IMS System, International Journal of Emerging Trends in Engineering Research, Volume 8, No. 1, Page, 88-113, 2020

[31] J. K. R. Sastry, K. Sai Abhigna, R. Samuel and D. B. K. Kamesh, Architectural models for fault tolerance within clouds at the infrastructure level, ARPN Journal of Engineering and Applied Sciences, VOL. 12, NO. 11, 2017, Pages 3463-3469

[32] DBK Kamesh, JKRSastry, Ch. Devi Anusha, P. Padmini, G. Siva Anjaneyulu, Building Fault Tolerance within Clouds at Network Level, International Journal of Electrical and Computer Engineering (IJECE), Vol. 6, No. 4, pp. 1560~1569, 2016 https://doi.org/10.11591/ijece.v6i4.10676

[33] S. L. SUSHMITHA, Dr. D. B. K. JKRSASTRY, V. V.ISHNA REDDY, building fault tolerance within clouds for providing uninterrupted Software as service, Journal of Theoretical and Applied Information Technology, Vol.88. No.1, Pages 65-76, 2016

[34] JKRSastry, M TrinathBasu, Securing Multi-tenancy systems through user spaces defined within the database level, Jour of Adv Research in Dynamical & Control Systems, Volume 10, issue 7, Page405-412, 2018

[35] JKRSastry, M TrinathBasu, Securing Multi-tenancy systems through multi DB instances and multiple databases on different physical servers, International Journal of Electrical and ComputerEngineering (IJECE), Volume 9, Issue 2, Pages 1385-1392, 2019. https://doi.org/10.11591/ijece.v9i2.pp1385-1392

[36] JKRSastry, M TrinathBasu, Securing SAAS service under cloud computing-based multi-tenancy systems, Indonesian Journal of Electrical Engineering and Computer Science, Volume 13, Issue 1, Page65-71, 2019 https://doi.org/10.11591/ijeecs.v13.i1.pp65-71

[37] M TrinathBasu, JKRSastry, Enhancing Data Security under Multi-Tenancy within OpenStack, International Journal of Advanced Trends in Computer Science and Engineering, Volume 9, Issue 1, 2020, pp.533-544

[38] Dr.JKRSastry, M. TrinathBasu, Enhancement of Security within OpenStack – Some measures, International Journal of Emerging Trends and Engineering Research, Volume 8, Issue 3, 2020,pp. 919-938

[39] Dr.JKRSastry, B. TrinathBasu, Implementing User defined Attribute and Policy based Access Control, International Journal of Emerging trends in Engineering Research, Volume 8. No. 7, July 2020, https://doi.org/10.30534/ijeter/2020/171872020

[40] *Panda B., TripathyS.N.,*Morphological malware detection: An API sequence mining with LCS based voting algorithm, Journal of Advanced Research in Dynamical and Control Systems, 2018, vol. 10, Iss. 6, pp. 625-623

[41] Venkateswara Rao V.M., Anand Kumar A.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

[42] Patil R., Prakash V.C. Neural network based approach for improving combinatorial coverage in combinatorial testing approach, Journal of Theoretical and Applied Information Technology,Vol. 96, iss. 20, pp 6677-6687

[43] Arshad M., Hussain A., A Novel multi-level attack detection and prevention model for Dynamic LAN/WLAN networks. RevistaTecnica de la Facultad de Ingenieria Universidad del Zulia, vol 41, iss. 1, pp 59-66

[44] Cheerla S., VenkataRatnam D., Teja Sri K.S., Sahithi P.S., Sowdamini G., Neural network based indoor localization using Wi-Fi received signal strength, Journal of Advanced Research in Dynamical and Control Systems, 2018, Vol. 10, Iss.4, pp.374-378

[45] Vara Prasad P.V., Sowmya N., Rajasekhar Reddy K., Jayant Bala P., Introduction to dynamic malware analysis for cyber intelligence and forensics, International Journal of Mechanical Engineering and Technology, Vol. 9, Iss. 1, pp 10-21

[46] SasiBhanu J., Lakshmi Prasad M., SastryJ.K.R. Combinatorial neural network based a testing of an embedded system, Journal of Advanced Research in Dynamical and Control Systems, 2018, Vol. 10, Iss. 7, pp 605-611

[47] Rao G.A., Syamala K., Kishore P.V.V., Sastry A.S.C.S., Deep convolutional neural networks for sign language recognition, Conference on Signal Processing And Communication Engineering Systems, SPACES 2018, pp 194-197
, 2018, Vol 10, Iss. 14, pp. 609-617

[48] Reddy A.V.N., Phani Krishna C., A survey on applications and performance of deep convolution neural network architecture for image segmentation disease classification from MRI images, Journal of Advanced Research in Dynamical and Control Systems