International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 05 | May -2017 www.irjet.net p-ISSN: 2395-0072

Review of Malware Data Classification and Detection in Smart Devices

Sandeep Sharma
Dept. of CS & Engineering
LNCTE, Bhopal (M.P.)
ssharma1886@gmail.com

Shrish Dixit
Dept. of CS & Engineering
LNCTE, Bhopal (M.P.)
shrilnct@gmail.com

Babita Pathik
Dept. of CS & Engineering
LNCTE, Bhopal (M.P.)
babitapathik@gmail.com

Dr. Shiv K. Sahu
Dept. of CS & Engineering
LNCTE, Bhopal (M.P.)
shivksahu@rediffmail.com

Abstract - The increasing rate of popularity of android based smart phone is day to day. The uses of smart phone compromise with various malware and infected virus. The malware and infected software degraded the performance of smart phone and android based system. The process of malware in smart device also theft the secured information and data over the third party. In this paper present the review of malware detection and classification based on different feature extraction and classification technique. The feature extractions play an important role in malware detection and classification. For the extraction of features used various data oriented features extractor.

***Keywords*: Smart devices, malware, android, classification**

1. INTRODUCTION

Smart devices are rapidly emerging as popular appliances with increasingly powerful computing, networking and sensing capabilities. Perhaps the most successful examples of such devices so far are smart phones and tablets, which in their current generation are far more powerful than early personal computers (PCs). The key difference between such smart devices and traditional "non-smart" appliances is that they offer the possibility to easily incorporate third-party applications through online markets. The popularity of smart devices –intimately related to the rise of cloud-computing paradigms giving complementary storage and computing services is backed by recent commercial surveys, showing that they will very soon outsell the number of PCs worldwide. For example, the number of smart phone users has rapidly increased over the past few years. Smart devices present greater security and privacy issues to users than traditional PCs. For instance, many of such devices incorporate numerous sensors that could leak highly sensitive information about users' location, gestures, moves and other physical activities, as well as recording audio, pictures and video from their surroundings. Furthermore, users are increasingly embedding authentication credentials into their devices, as well as making use of on-platform micropayment technologies such as NFC. SYMBIAN OS security model is based on a basic permission system. Phone resources are controlled by the OS using a set of permissions called capabilities [4]. Furthermore, applications run in user space, while the OS run in kernel space. Those applications requiring access to protected libraries must be signed using a certificate issued by Symbian, while all others can be self-signed. Protection at the market level is inexistent or very low. BLACKBERRY security model is based on a coarse-grained permission protection model. Applications have very limited access to the device resources and, as in the case of BLACKBERRY OS, they must be signed by the manufacturer (RIM) to be able to access resources such as, for example, the user's personal information. Additionally, applications must get user authorization to access resources such as the network. However, once the user grants access to an application to use the network, the application can both send SMSs and connect to Internet. Although applications are not executed in a sandbox, some basic process and memory protection is offered. For instance, a process cannot kill other processes nor access memory outside the app bounds. Google's ANDROID OS security model relies on platform protection mechanism rather than on market protection, as users are free to download applications from any market. Applications declare the permissions they request at installation time through the so-called manifest. If the user accepts them, the operating system will be in charge of enforcing them at running time. Apples IOS security model relies on market protection mechanisms rather than enforcing complex per-mission polices on the device at installation time. Apple's App Store is a walled-garden market with a rigorous review process [7]. Those processes are essential for preventing malware from entering the device, as runtime security mechanisms are limited to sandboxing and user supervision. IOS isolates each third-party application in a sandbox. However, most of the device's resources are accesible1 and misuse of a few of such as GPS, SMS, and phone calls can only be detected by the user after installation. Furthermore, IOS sandboxing model is weaker than ANDROID OS's, as Apple only uses one sandbox to run all applications, whereas Google separates each application in a sandbox. Microsoft's market protection model for WINDOWS MOBILE systems is based on application review. Developers are also validated prior to application's approval. Platform protection in WINDOWS MOBILE is similar to ANDROID OS. It uses a trusted boot component and code signing to protect the integrity of the operating system. It also provides signed drivers and applications through the Windows Phone Store online market. Malware software that

exhibits malicious behavior is broadly categorized to include viruses, botnets, worms, and Trojan horses. Initially, malware merely highlighted a software system's security vulnerabilities, but the motivations behind it gradually changed, and its authors now use malware to gain financial benefits on a larger scale [10]. There are three different types of malware detection techniques: attack or invasion detection, misuse detection (signature-based) and anomaly detection (behavior-based). Attack or Invasion detection tries to detect unauthorized access by outsiders. But, misuse detection (signature-based) tries to detect misuse by insiders and describes very good detection results for specified, well-known attacks. The advantages of misuse detection are: it has no false positives and can quickly detect intrusion. Disadvantage is not capable of detecting new unfamiliar intrusions, even if they are built as minimum variants of already known attacks. Anomaly detection (behavior-based) refers to detecting patterns in a given dataset that do not conform to an established normal behavior. It also attempts to estimate the abnormal behavior of the system to be protected and generate anomaly alarm whenever the deviation between a given observation at an instance and normal behavior exceeds a predefined threshold [12]. Advantage is potential to detect previously unseen intrusion events and disadvantage is many false positives and requires a large set of training data to construct normal behavior profile. For removing these shortcomings of misuse detection and anomaly detection profiles should be updated with large amount the datasets at regular interval of time.

2. Related work

In this section discuss the related work in malware detection in smart devices. Various authors and researcher used various technique here discuss some technique along with their title.

Et al.	Author	Title	Approach	Publication
[1]	Ke Xu, Yingjiu Li and Robert H. Deng	ICCDetector: ICC-Based Malware Detection on Android	They talked about another malware recognition strategy, named ICCDetector. ICCDetector yields a location display in the wake of preparing with an arrangement of amiable applications and an arrangement of malwares, and utilizes the prepared model for malware discovery. The execution of ICCDetector is superior to the benchmark in their trials. The malwares identified by ICCDetector are classified into five new malware classifications as per their ICC attributes, which clarifies the connection between malware practices and ICC designs.	IEEE, 2016
[2]	Guillermo Suarez-Tangil, Juan E. Tapiador, Flavio Lombardi and Roberto Di Pietro	ALTERDROID: Differential Fault Analysis of Obfuscated Smartphone Malware	They have displayed ALTERDROID , a structure for malware investigation in light of the thought of differential blame examination. They have depicted its engineering and star vided a formal model of differential blame examination. Also, they have displayed an open-source model usage of ALTERDROID with a flexible outline that can be the reason for further research here. Differential blame examination in the route actualized by ALTERDROID is an intense and novel element investigation system that can distinguish conceivably noxious	IEEE, 2016

			components covered up inside an application bundle.	
[3]	Luca Caviglione, Mauro Gaggero, Jean-François Lalande, Wojciech Mazurczyk and Marcin Urbanski	Seeing the Unseen: Revealing Mobile Malware Hidden Communications via Energy Consumption and Artificial Intelligence	They have concentrated on the intriguing application situation, which is portrayed by two procedures attempting to impart outside their sandboxes for noxious purposes, for example, for delicate information exfiltration. Two recognition strategies have been produced, requiring the arrangement of relapse and classification issues. To check their adequacy, they have executed seven nearby secretive channels on the Android stage, and they have played out a trial estimation and location crusade. The acquired outcomes demonstrate that both techniques are portrayed by a decent discovery execution and can be utilized as a precise IDS programming on a present day cell phone to uncover the nearness of perils abusing data stowing away.	IEEE, 2016
[4]	Lilian D. Coronado-De-Alba, Abraham Rodríguez-Mota and Ponciano J. Escamilla-Ambrosio	Feature Selection and Ensemble of Classifiers for Android Malware Detection	Every one of the calculations was connected utilizing their default parameters in Weka for binarized traits and ostensible class so as to acquire a model in light of a meta-learner. The accuracy of every individual calculation was taken as the principle trademark keeping in mind the end goal to pick the best mix of choices. The element determination was made utilizing Chi-Square and Relief demonstrating that it is not an applicable exercise since similar outcomes were gotten when no element choice was performed.	IEEE, 2016
[5]	Yu Feng, Saswat Anand, Isil Dillig and Alex Aiken	Apposcopy: Semantics-Based Detection of Android Malware through Static Analysis	They displayed Apposcopy, a static investigation approach for recognizing malware in the versatile applications biological community. Apposcopy performs profound static investigation to concentrate information flow and control-flow properties of Android applications and utilizations these outcomes to recognize whether a given application has a place with a known malware family. Their analyses demonstrate that Apposcopy can distinguish malware with high exactness and that its marks are flexible to different	ACM, 2014

			program confusions.	
[6]	Prof. Amruta Gadekar, Sharad Goykar, Shesharao Chatse and Vishaka Deore	A Survey on a ICC-Based Malware Detection on Android	They have executed seven nearby secretive channels on the Android stage, and they have played out a trial estimation and location crusade. The acquired outcomes demonstrate that both techniques are portrayed by a decent discovery execution and can be utilized as a precise IDS programming on a present day cell phone to uncover the nearness of perils abusing data stowing away.	IJETCS, 2016
[7]	Suleiman Y. Yerima, Sakir Sezer, Gavin McWilliams and Igor Muttik	A New Android Malware Detection Approach Using Bayesian Classification	They displayed a meta-troupe calculation for malware identification in Android applications applying static investigation for recognizing vindictive applications. The strategy utilized distinctive datasets with adjusted and uneven number of generous and malware tests, with or without highlight determination. Datasets were acquired from the Drebin extend for malware tests and the Google Pay Store and outsider stores.	IEEE, 2013
[8]	Mingshen Sun, Xiaolei Li, John C.S. Lui, Richard T.B. Ma and Zhenkai Liang	Monet: A User-oriented Behavior-based Malware Variants Detection System for Android	They show the plan and usage of MONET to recognize malware variations and to protect against change assault. MONET will create a runtime conduct signature which comprises of RBG and SSS to precisely speak to the runtime conduct of a malware. Their framework incorporates a backend recognition server and a customer application which is anything but difficult to convey on cell phones.	arXiv, 2016
[9]	Hugo Gascon, Fabian Yamaguchi, Daniel Arp and Konrad Rieck	Structural Detection of Android Malware using Embedded Call Graphs	They have introduced a learning-based technique for the identification of pernicious Android applications. Their technique utilizes an express element outline by the area hash diagram bit to speak to applications in light of their capacity call charts. This portrayal is appeared to be both, efficient and effective, for preparing a SVM that at last empowers us to naturally distinguish Android malware with a location rate of 89% with 1% false positives, comparing to one false caution in 100 introduced applications on a cell phone.	ACM, 2013

[10]	Borja Sanz, Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero, Javier Nieves, Pablo G.Bringas and Gonzalo Álvarez	MAMA: Manifest Analysis for Malware Detection in Android	They assessed the limit of these two capabilities to recognize malware utilizing machine-learning methods. To approve their strategy, they gathered malware and kindhearted specimens of Android applications. At that point, they separated the previously mentioned highlights for every application and prepared the models, demonstrating that the blend of these two components can give high precision distinguishing malware. All things considered, there are a few contemplations in regards to the suitability of their approach.	ACM, 2013
[11]	Pengbin Feng, Jianfeng Ma and Cong Sun	Selecting Critical Data Flows in Android Applications for Abnormal Behavior Detection	They display an approach SCDFLOW, which can choose basic information flows and take these flows as components to recognize malware in light of strange information flows. The novel calculation CFlowSel talked about by SCDFLOW can choose basic information flows in light of their event frequencies between considerate applications and malware. Through investigations, they confirm that CFlowSel beats two existing component determination calculations. They likewise demonstrate that this calculation can viably lessen the number of information flow highlights for unusual location on the two datasets. Evacuating insignificant and uproarious information flows and considering basic information flows as elements will enhance the accuracy of malware identification in view of anomalous information flows. SCDFLOW, contrasted and MUDFLOW, successfully enhances the malware location rate by 5.73% on dataset MW and 9.07% on dataset DN and causes unimportant increment on memory utilization.	Springer, 2017
[12]	Yousra Aafer, Wenliang Du and Heng Yin	DroidAPIMiner: Mining API-Level Features for Robust Malware Detection in Android	They have statically examined a vast corpus of Android malwares having a place with different families and a huge kindhearted set having a place with different classifications. They have directed a recurrence examination to catch the most significant API calls that malware summon, and refined the list of capabilities to	Springer, 2014

			prohibit API calls made by outsider bundles. They played out a straightforward information flow examination to get unsafe contribution to a few API calls.	
[13]	Borja Sanz, Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero, Pablo Garcia Bringas and Gonzalo ´Alvarez	PUMA: Permission Usage to detect Malware in Android	They exhibit PUMA, another strategy for distinguishing vindictive Android applications through machine-learning systems by breaking down the extricated authorizations from the application itself. They assess the limit of authorizations to distinguish malware utilizing machine-learning strategies. So as to approve their technique, they gathered 239 malware tests of A droid applications.	Springer, 2013
[14]	Nirmala Yadav,Aditi Sharma and Amit Doegar	A Survey on Android Malware Detection	The developing rate of Android malware made a trouble in life of Android clients. Client feels uncertain as with hazard like hanging of telephone on getting a call, individual data taking and huge measure of bill while interfacing with web and some more. The accessible Android malware identification approaches has not possessed the capacity to give better precision.	IJNTR, 2016
[15]	Geethu M Purushothaman, G Gopinadh and Nihar SNG Sreepada	Malware Detection in Android	They have made a system which recognizes Android.bgserv malware. This application naturally switches on the Bluetooth and Wi-Fi administrations of the framework and makes the framework helpless against assaults. Their framework on recognition showcases to the client a ready delivery person which demonstrates the nearness of malware and powers the client to uninstall the application.	IJARCET, 2014

3. MALWARE CATEGORIES

Malware forms a role of verity of software application. It activated may be on desktops, servers, mobile phones, printers, and programmable electronic circuits. Sophisticated attacks have confirmed data can be stolen through well written malware residing only in system memory without leaving any footprint in the form of persistent data. Malware has been known to disable information security protection mechanisms such as desktop firewalls and anti-virus programs.

Some even have the ability to subvert authentication, authorization, and audit functions. It has configured initialization files to maintain persistence even after an infected system is rebooted. Upon execution, sophisticated malware may self-replicate and/or lie dormant until summoned via its command features to extract data or erase files. Here we describe categories of malware software in from of table.

Malware categories	Propagation	Infection	Self-Defence	Capabilities
Key logger	Infected websites and/or USB or other media	Vulnerable browsers or unpatched OS or application	Replace IO device drivers or APIs	Collect user keystrokes including credentials
Rootkit	Infected websites and/or installs on servers by hackers or insiders	Exploited trusted admin access, vulnerable browsers, or unpatched OS or application	Replacing OS kernel-level API routines	Collect data and impersonate user activity for entire machine and its interfaces
Flaw Exploits	Execution of unexpected commands to flawed software by remote hackers	Vulnerable software-to-database and command execution interfaces	Impersonation of authorized users	Download or upload data from data repositories between target and malware operator site
Bot	Bots are generally delivered via infected websites, or links to malicious websites embedded in phishing email.	User may voluntarily install individual bots based on deceptive messages in email or web instruction, or via browser/OS vulnerabilities.	Bot updates security patches and anti-virus on machine to ensure stable operation and keep other bots out. Lays dormant until activated.	When activated by botnet operator, the operator may direct bot to execute a variety of standard or custom functions.
Denial of Service (DOS)	IP packet delivery	Internet protocols that automate packet processing	Simultaneously attack from multiple sources	Consume computing resources on targets

Table 2 refers only to single pieces of software and that there is no hierarchy in malware classification. However, alluded to in the description of a bot is the fact that a typical cybercrime will require multiple different types of software acting in coordination in order to achieve the full crime capability. For example, a criminal may use email spamming software (a form of flaw exploit) to trick a user into downloading a key logger from an infected website. The criminal would then have to host a site for the key logger to deliver the stolen credentials.

4. CONCLUSIONS

In this paper, have evaluated and dissected the current malware identification strategies and contrast it and the favorable position and inconvenience furthermore talk about some present issue are remain. From the investigation scientist, has centered another dynamic component extraction of malware discovery systems. Some procedure of method considering govern mining and some other are based self-spread element extraction strategy. This will contribute thoughts in malware location strategy field by creating an enhance technique for malware recognition. Additionally, although crime ware and state-supported digital assaults and battles are the most noticeable type of assault, FIs ought to perceive the expanding risk from both outer and inside sources, and take pragmatic measures to recognize and protect against potential interior malware impedance with business handle. Is ought to assess their defenselessness to the malware depicted in this paper.

REFERENCES

[1] Ke Xu, Yingjiu Li and Robert H. Deng "ICCDetector: ICC-Based Malware Detection on Android", IEEE, 2016, Pp 1252-1264.

[2] Guillermo Suarez-Tangil, Juan E. Tapiador, Flavio Lombardi and Roberto Di Pietro "ALTERDROID: Differential Fault Analysis of Obfuscated Smartphone Malware", IEEE, 2016, Pp 789-802.

[3] Luca Caviglione, Mauro Gaggero, Jean-François Lalande, Wojciech Mazurczyk and Marcin Urbanski "Seeing the Unseen: Revealing Mobile Malware Hidden Communications via Energy Consumption and Artificial Intelligence", IEEE, 2016, Pp 799-810.

[4] Lilian D. Coronado-De-Alba, Abraham Rodríguez-Mota and Ponciano J. Escamilla- Ambrosio "Feature Selection and Ensemble of Classifiers for Android Malware Detection", IEEE, 2016, Pp 1-6.

[5] Yu Feng, Saswat Anand, Isil Dillig and Alex Aiken "Apposcopy: Semantics-Based Detection of Android Malware through Static Analysis", ACM, 2014, Pp 1-12.

[6] Prof. Amruta Gadekar, Sharad Goykar, Shesharao Chatse and Vishaka Deore "A Survey on a ICC-Based Malware Detection on Android", IJETCS, 2016, Pp 1-5.

[7] Suleiman Y. Yerima, Sakir Sezer, Gavin McWilliams and Igor Muttik "A New Android Malware Detection Approach Using Bayesian Classification", IEEE, 2013, Pp 1-8.

[8] Mingshen Sun, Xiaolei Li, John C.S. Lui, Richard T.B. Ma and Zhenkai Liang "Monet: A User-oriented Behavior-based Malware Variants Detection System for Android", arXiv, 2016, Pp 1-13.

[9] Hugo Gascon, Fabian Yamaguchi, Daniel Arp and Konrad Rieck "Structural Detection of Android Malware using Embedded Call Graphs", ACM, 2013, Pp 1-10.

[10] Borja Sanz, Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero, Javier Nieves, Pablo G.Bringas and Gonzalo Álvarez "MAMA: Manifest Analysis for Malware Detection in Android", ACM, 2013, Pp 1-19.

[11] Pengbin Feng, Jianfeng Ma and Cong Sun "Selecting Critical Data Flows in Android Applications for Abnormal Behavior Detection", Springer, 2017, Pp 1-14.

[12] Yousra Aafer, Wenliang Du and Heng Yin "DroidAPIMiner: Mining API-Level Features for Robust Malware Detection in Android", Springer, 2014, Pp 1-18.

[13] Borja Sanz, Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero, Pablo Garcia Bringas and Gonzalo ´Alvarez "PUMA: Permission Usage to detect Malware in Android", Springer, 2013, Pp 1-10.

[14] Nirmala Yadav,Aditi Sharma and Amit Doegar "A Survey on Android Malware Detection", IJNTR, 2016, Pp 47-53.

Geethu M Purushothaman, G Gopinadh and Nihar SNG Sreepada "Malware Detection in Android", IJARCET, 2014, Pp 1429-1436.