APP MISBEHAVIOUR CHECK: DEVELOPMENT OF VIRUS MODELING, PROPAGATION & DETECTION USING ANDROID

S. Anushya[1], A. Gayashri[2], S. Ebenazer Roselin[3], S. Shalini[4]

[1,2]UG Scholar, Department of Computer Science and Engineering, Prince Dr. K. Vasudevan College of Engineering and Technology, Chennai, India.
[3,4] Assistant Professorsl, Department of Computer Science and Engineering, Prince Dr. K. Vasudevan College of Engineering and Technology, Chennai, India.

Abstract- *Users are affected by zombie app and the important issues of mobile devices are energy consumption. For that we implement, facilitate affective identification and subsequent quarantine of such zombie apps towards stopping their undesired activities. Initially we used to design multiple application and create an antivirus like application which we implement and to monitor. The common virus attacking behaviours like sending SMS, Drain of battery, storage of call logs in cloud server, crashing the gallery and also the mobile get hanged by sending RAM functionalities. All these misbehaviour activities are monitored and quarantined successfully with the user acknowledgement by our application. We implement the effective technique which will compute the virus behaviour in mobile and also we present the antivirus to overcome the misbehaviour app.*

Keywords-Android, quarantine, misbehaviour, monitored, antivirus

1. INTRODUCTION

It is ability to compute while on the move and access information from anywhere or anytime. It has two key factors mobility is a capability to change a location and computing is a capability to automatically carried out certain process related to service investigation.

In most of the applications, design is that the area is strongly driven by innovation, characterised by rapidly evolving use, and has enormous market potential and growth. New technologies are constantly being developed, new use domain are constantly being explored, and successful new ideas and applications reach millions of user. In fact, by the end of 2010 more smart-phones than personal computers were for the first time, being sold worldwide, with more than 100 million units shipped in the last three months of that year alone.

During its lifetime, it has expanded from being primarily technical to now also being about usability, usefulness, and user experience. This has led to the birth of the vibrant area of mobile interaction design at the intersections between, among others, mobile computing, social sciences, human-computer interacting, industrial design, and user experience design. Mobile computing is a significant contributor to the pervasiveness of computing resources in modern western civilisation. In concert with the proliferation of stationary and embedded computer technology, throughout society, mobile devices such as cell phones and other handheld or wearable computing technologies have create a state of ubiquitous and pervasive computing where we are surrounded by more computational devices than people(Weiser 1991).

The field of mobile computing has its origin in fortunate alignment of interests by technologists and consumers. Since the dawn of the computing age, there have always been technological aspirations to make computing hardware smaller, and ever since computers became widely accessible, there has been a huge interest from consumers in being able to bring them with you. As a result, the history of mobile computing is paved with countless commercially available devices.

The portability was about reducing the size of hardware to enable the creation of computers that could be physically moved around relatively easily. Miniaturization was about creating new and significantly smaller mobile from factors that allowed the use of personal mobile devices while on the move. Connectivity was about developing devices and applications that allowed users to be online and communicate via wireless data networks while on the move. Convergence was about integrating emerging types of digital mobile devices, such as Personal Digital Assistants (PDAs), mobile phones, music players, cameras, games, etc., into hybrid devices. Divergence took an opposite approach to interaction design by promoting information appliances with specialised functionality rather than generalized ones. The latest wave of apps is about developing matter and substance for use and consumption on mobile devices, and making access to this fun or functional interactive application content easy and enjoyable. Finally, the emerging wave of digital ecosystems is about the larger wholes of pervasive and interrelated technologies that interactive mobile systems are increasingly becoming a part of it.

We use a SAAS (Software as a service) concept, it is on-demand software and it is a method deliver service over the internet, applications accessed via internet. The applications run on SAAS providers servers provide manager security, availability and perform.

Unused application would do background activities have significant negative impacts on the user, e.g., leaking private information or significantly taxing resources such as the battery or network and to identify such unused application, which we call as zombies. There is no system to deactivate these zombie apps. Zombie apps will access the user private

information. We the designed a zap-droid, it is an effective way to prevent the undesired activities and also zap-droid restores the app quickly and effectively. It is energy efficient consuming 3% of the battery per day.

We have some challenges first we need and efficient mechanism to track the status of the application, Zombies apps are active in the background, the approaches can be resource intensive. Second, it will monitor the application use sensitive resource protected by permission in a lightweight manner and android does not allow the one application to track the permission access patters to other application. Third, the application that will not use for prolong period is quarantined. Fourth, previously-quarantined zombie app is restored (reinstall from Google Play Store).

Most of the people will download the app more than once and 20% of user will never delete the downloaded apps (i.e., user will not use them for prolong time or periods). In our applications, it will monitor the apps that not use for prolong period will automatically requesting the user for uninstalling or quarantine the apps.

2. RELATED WORK

To identify and to understand the zombies apps analyze the application in which the zombies are affected and the apps in which battery is drained. The apps which are doing malicious activities are identified and restored. In this, concept the apps doing malicious activities are understood and identified.[1]we present the profiling android applications that was lacking the user interaction. Apps can communicate with many sources than users. Then the profile about the android applications which is used are named as Profile Droid, which creates the multiple layers that are monitoring the behaviour characters of the apps are use[2]. The android application is metered in smart phones using kernel activity monitoring .In real time the online and autonomous estimation in analyzing the battery .The fine-grained energy consumption information is portable in the system. The apps is easy to use and metering battery percentage. By reading the hardware performance in the counter measures [3].A system is to new advertising API and corresponding advertising permission for the android platform is called Ad-Droid .The android permissions can be advertisers such as AdMob and Millennial Media play a key role in the ecosystem. The allowing advertising in the smart phones it sends acknowledgement to users in the system. The System functionality are GET_TASKS and VIBRATE. The privilege separation for applications and advertisers in android is advertising [4] .In this concept the apps which are performing in the smart phones that are maintaining our energy level. That which app was draining our battery level and performance was monitoring in the wild. The mobile that are identify the critical path in the user transactions. The App Insight was guided by three principles that are low overhead, zero effort, immediately deployable. Our analysis is to perform app in the execution process[5]. The android applications for ecosystem that analysis automatically which apps are doing malicious activities in device. The system is also prefers the security and privacy. Refactoring the binary applications packages according to user's security preferences. In this android system the application security is governed by digital signatures and a list of coarse grained application are needed [6].

In this application the smart phones are commonly most of the users used only the android applications smart phones. The application which are downloaded in the Google play store by users that was noticed and the smart phones are consuming the energy. The apps which are energy consumption is high that are monitored by the applications which we are used. That we are developing a smart energy for monitoring the system called SEMO. It can profile the mobile application of battery usage and vital for both users and the developers [7]. Android allows the third party applications because it is an open source platform. It with an extensive application that are access to phone hardware and settings are change. Once the users have to install android app that gives permission by acknowledgement and to agree or disagree is needed to use the app. This is included the privacy and security from the application [8]. In this consuming the energy from the smart phones are need to analysed. This approach is to power consumption of the battery usage analysis in the system. The smart phones that pocket sized that draining the high battery level. So this system is to analysis the energy consumption in the modern mobile devices. [9] Normally the embedded device will consume high energy perspectives but also should produce to save the less power. Open handset alliance hosting members released the open source platform android mobile devices. The android mobile device can be optimized and implementing in Dalvik JVM [10]. This paper describes about system gets down the power booter can be used in built-in-memory. That are power booter is the automated power model that are used.

3. EXISTING SYSTEM

Understanding zombie apps will measure the unwanted behaviours of candidate zombie apps. The Amazon's Mechanical Turk will identify the apps that are not used for month-long period and understand the behaviour of the particular application.

Identifying and profiling the zombie apps are like finding unused apps, detecting foreground apps, lightweight implementation, measuring resource consumption monitoring network usage and CPU usage, determining battery consumption and storage, tracking apps permission access patterns and permissions invoked by unused apps [2]. Quarantining zombie apps is to design the module of zap-droid for revoke permissions from zombie apps. It can be categorized as likely to restore and unlikely to restore. First category, the relevant data's are stored on device itself and second category, the data's is moved to either cloud or other device. Restoring zombie apps: The quarantined app will be restored from device or cloud.

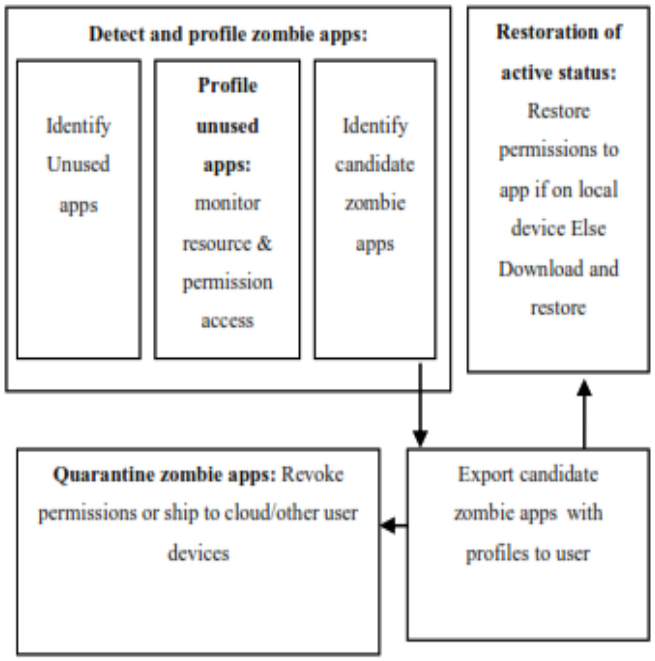

Fig 1. High-level Architecture of zap-droid

4. PROPOSED SYSTEM

In our implementation; first of all we need to design multiple applications. We also create an Anti virus like Application so as to monitor. The virus behaviours are like Sending SMS, Storage of call logs in the Cloud server, Crashing the Gallery, Drain of Battery, Slowness of Mobile by reducing the RAM. All these misbehaviour activities are monitored and quarantined successfully with the acknowledgement of Application [5]. This application is only used for Android users [4]. If any application is draining the battery level those application will shows on user mobile and it will be automatically uninstalled from user mobile and send it to recycle bin. User can install that application wherever they need.

We seek to facilitate effective identification and subsequent quarantine of such zombie apps toward stopping their undesired activities. We are creating both virus and antivirus. For virus creation, first if the user will have some call means it will show the call is infected by virus, sample folder created, wake lock on and service enabled. Second the viruses are crashing the gallery.

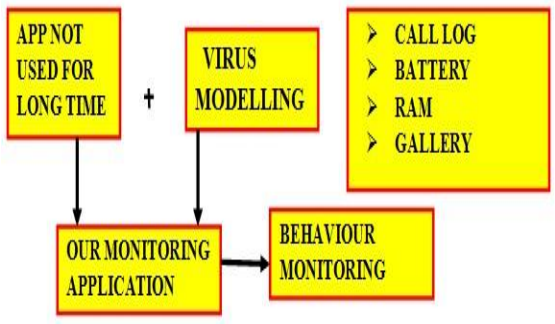

Fig.2.Block Diagram of App Misbehaviour Check

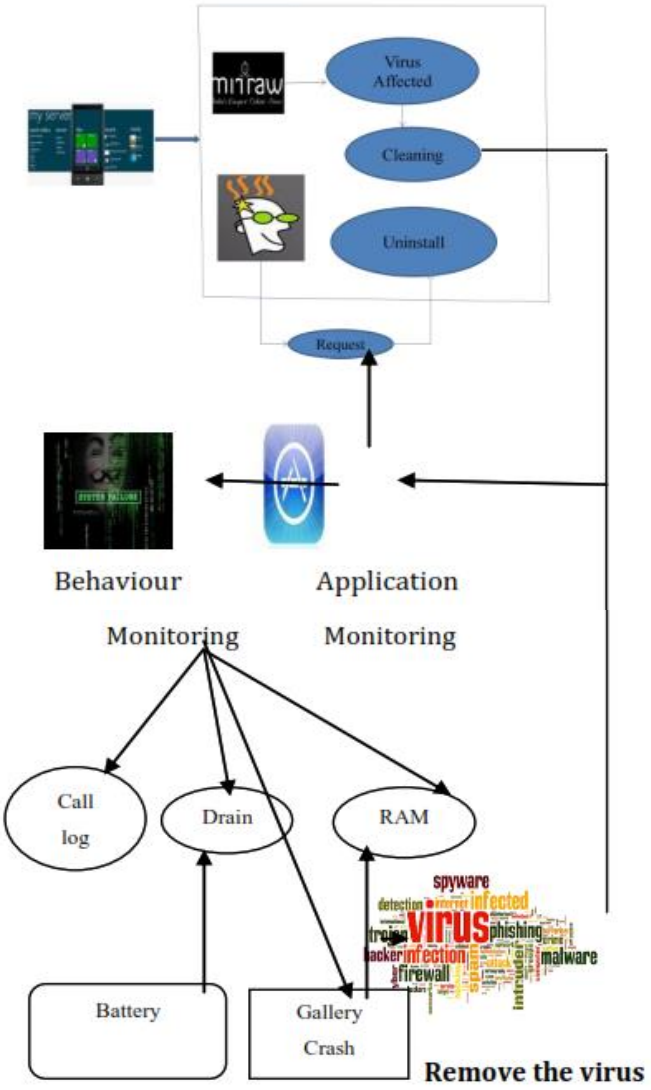

Fig.3. System architecture for App misbehaviour check

Third, if user will receive some messages means it will automatically send a reply messages to receiver. In our application, monitoring the behaviours of all apps and unused apps are quarantined. The user need not to update the application for each time, if the server will update means it will automatically done the process. It has constant memory and the batch files are stored in cloud itself.

Since a user can change her mind about whether or not she wants to use an app, a zombie app must be restorable as quickly as possible if the user so chooses [6]. After an app goes unused by a user for a prolonged period, the determination of whether the app should be constrained depends on whether the app's resource usage during the period of unused is considered significant or whether the app's access of private data is deemed serious. Moreover, the manner in which a zombie app should be quarantined depends on whether the user is likely to want to use the app again in the future (e.g., a gaming app that the user tried once and decided is not interesting versus a VoIP app that the user uses infrequently).

Avoid Data or information leakage. Reliable-app consistently good in quality or performance and it can be trusted to work well. High data transmission data's over a point to point or point to multipoint communication channels. It will provide antivirus for designed to detect and destroy computer viruses or other virus.

1. ANDROID DEPLOYMENT:

Mobile Client is an Android application which created and installed in the User's Android Mobile Phone. So that we can perform the activities of the Application First Page Consist of the User registration Process. We'll create the User Login Page by Button and Text Field Class in the Android.

While creating the Android Application, we have to design the page by dragging the tools like Button, Text field, and Radio Button. Once we designed the page we have to write the codes for each. Once we create the full mobile application, it will generated as Android Platform Kit (APK) file. This APK file will be installed in the User's Mobile Phone an Application.

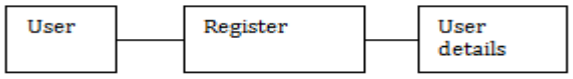

Fig.4. Android deployment

2. SERVER:

The Server is Server Application which is used to communicate with the Mobile Clients. The Server can communicate with their Mobile Client by GPS Technology. The Server Application can be created using Java Programming Languages.

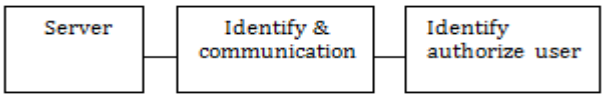

Fig.5. Server

3. MODELING VIRUS:

In this Module we will create the Mobile Virus which is malicious code that will perform malicious activities in the User's Mobile Phones. In this Project we are creating a New Folder Virus which will create a Folder inside the Folder virus by developing malicious codes so that we can generate the Mobile Virus. Once the attackers created the Virus, they will spread the Virus via SMS technique, So that the virus will be spread to other Users Mobile Phones. While sending via Bluetooth (Sharing Your Virus App) technique, the User's has to be present within the communication range. The Attacker can send the virus file via Mobile Application that was installed in their Mobile Phones.

4. MONITORING CALL LOGS & CRASHING GALLERY:

In this module we will create the mobile virus and spread the mobiles, call logs are stored in cloud server like missed call and dialled calls (Recent Call Log). Also spread the gallery virus so all data's are changed in encrypted format. So does not view the gallery.

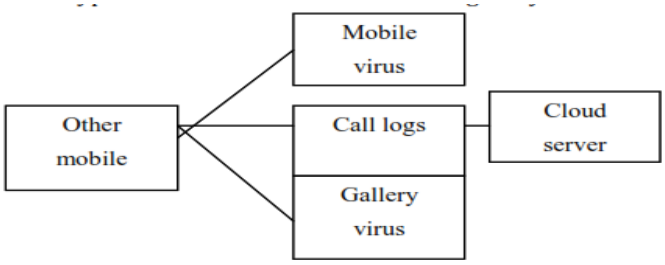

Fig.6. Monitoring call logs and crashing gallery

5. VIRUS PROPAGATION:

Once the attack spread the Virus File to other User's Mobile Phone the content of the message of the file will be analyzed by the Server to detect whether the file contains that Malicious Behaviour or not. If the file contains the malicious behaviour, then the Server will detect the file as Virus file. Once the Server detected that the Virus file it will deliver the patches to the User's Mobile Phone and deletes the Virus File.

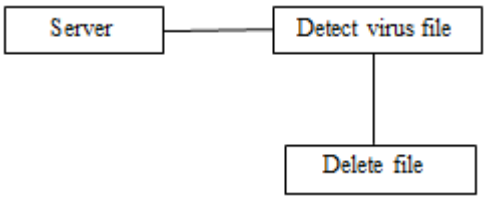

Fig.5. Virus propagation

5. DISTRIBUTION OF PATCHES:

Once the Server identify virus file was sent to the User's Mobile Phone, the Server will provide (SAAS) the patch files to delete the Virus file. Using an Android Application the patches will be distributed to the User's Mobile phone automatically to clear the Virus.

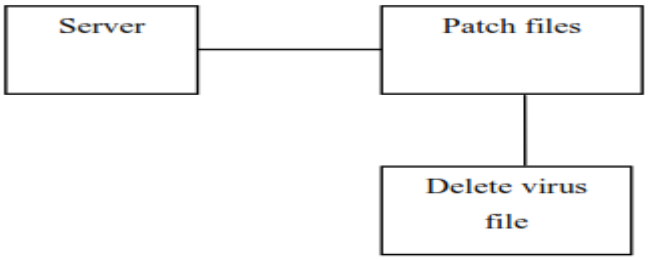

Fig.6. Distribution of patches

7. CLOUD & INFREQUENT APP ACCESS:

Cloud storage is a service model in which data is maintained, managed, backed up remotely and made available to users over a network. The contacts, IMEI number, IMSI number of mobile will be automatically backed up to cloud.

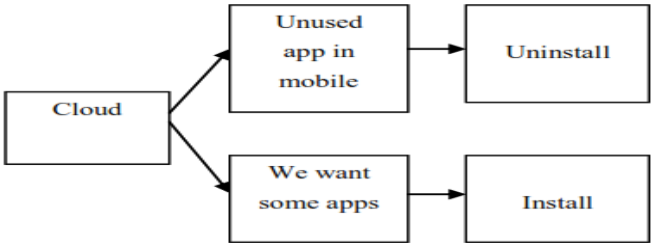

Fig.7. Cloud and infrequent app access

Another one is people are not using all apps in mobile so that mobile memory will be wasted for unwanted application. So we are showing that infrequent application to the user to uninstall it. If they give permission it will be uninstalled and through it to recycle bin whenever we want app we can install it from recycle bin.

5.METHODOLOGY

TIME-BASED SCANNING &MISBEHAVIOR MONITORING:

In this paper, we are creating a virus and antivirus. Misbehavior activities are done by virus that will be monitored by our application. The virus will do unwanted message to user , crashing the gallery, Slowness of mobile by reducing the RAM.For calls, at the time the user will notify that the virus is infected, service enabled, lock on activities are performed and also the sample folder is created.

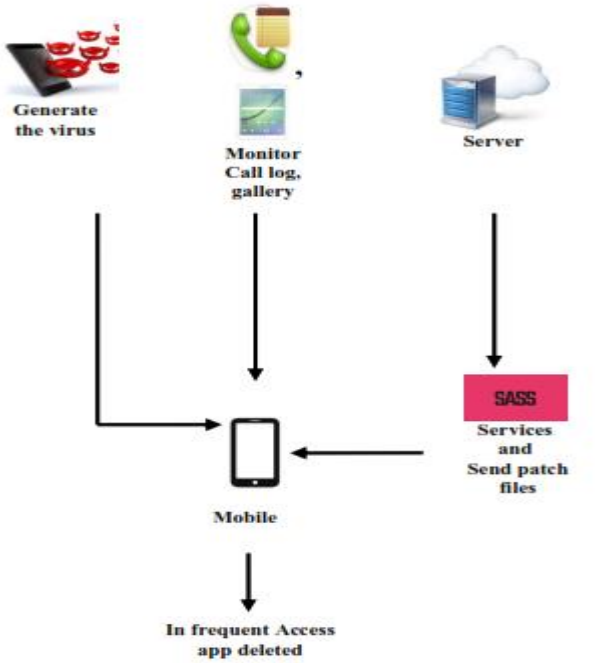

Fig.5.Monitoring misbehaviour activities and Time based scanning

In our application we create a antivirus so as to monitor, there is a constant memory. The call log will be stored in cloud itself. This application will monitoring the misbehavior activities of apps that are infected by virus. The unused application will be monitored and it will sent the request to our apps whether the application is uninstall or not and also the virus is affected means the cleaning process is started by our apps.

First, we are generate the virus to the mobile and they monitor call logs and gallery. From the server the sms process will take places services and send patch files to the mobile. In mobile, if they is infrequent access app will be deleted.

6. RESULT AND EXPERIMENT

For virus, the call log will stored in cloud itself and it will be stored the call no's and address of the users .The user receive the call from contact list or someone else at the time ,we have some notifications like this wake lock on, services are enabled, sample folder created and atlast it is infected . Next, the receiver wil the some message to the user by the infected virus it will send the message automatically to the receiver.

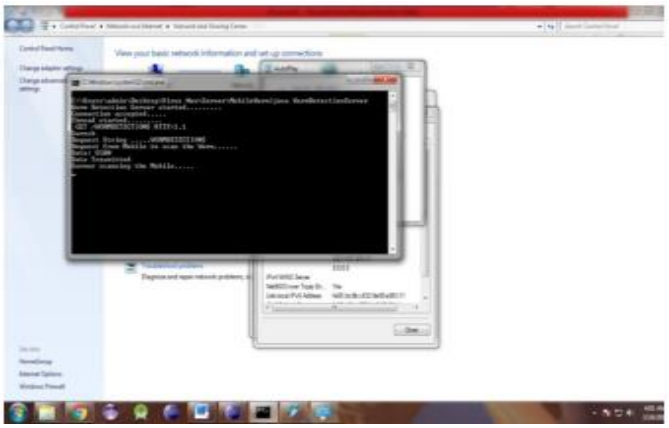

Fig.5. Server Started

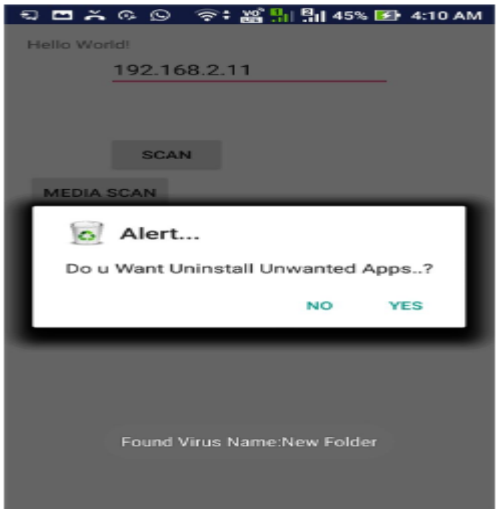

Fig.5. Monitored unwanted apps alert notification to the user.

If the messages are failed means it will send the message again and again to the receiver. Finally, the gallery will be crashed by virus and it slowness the battery.

The pin will be set and confirm the pin from the user the it is done successfully and the spam is created a file with content. Then create an account for the user, if the user will receive any call means the phone no, imei no, imsi no and the network will be stored in server itself.
The application will be scanning the app with the given server ip address then the apps will started to scanning if any virus if infected means it will clean the apps and send the alert message to user that the unwanted apps are to uninstall or not.

The alert message will be sender then the virus name will be found and created. Finally, scanning the apps successful and bool.

Fig.5. Scanning the apps with the server IP address

Starting from the end of 2011, attackers have increased their efforts toward Android smart phones and tablets, producing and distributing hundreds of malicious apps. These apps threaten the user data privacy, money and device integrity, and are difficult to detect since they apparently behave as genuine apps bringing no harm. This paper proposes MADAM, a multi-level host-based malware detector for Android devices. By analyzing and correlating several features at four different Android levels, MADAM is able to detect misbehaviors from malware behavioural classes that consider 125 existing malware families, which encompass most of the known malware. To the best of our knowledge, MADAM is the first system which aims at detecting and stopping at run-time any kind of malware, without focusing on a specific security threat, using a behaviour-based and multi-level approach. Not only the accuracy of the runtime detection of MADAM is very high, but it also achieves low performance (1.4%) and energy overhead (4%).

8. FUTURE ENHANCEMENT

In our paper create a malicious virus and monitor the virus attack. The malicious virus is automatically transmitting data through the SMS. The virus affect the gallery, call logs and slowness of RAM speed. To detect the virus attacks and delete the files through the patch files. The patch files are distributed via the cloud. In future patch jar files are stored in the AWS. When we need to clear the malicious files we access the patch files directly on the Amazon web services.

REFERENCES

1. Singh, S.V. Krishnamurthy, H. V. Madhyastha and I. Neamtiu ”Zap-Droid: Managing Infrequently Used Application OnSmartphones” IEEE Transaction on Mobile computing ,vol. 16, no. 5, pp1475-1489,MAY2017.

2. X. Wei, L. Gomez, I. Neamtiu, and M. Faloutsos, “Profile Droid: Multi-layer profiling of android applications,” in Proc. 18th Annu. Int. Conf. Mobile Comput. Netw., 2012, pp. 137–148.

3. C. Yoon, D. Kim, W. Jung, C. Kang, and H. Cha, “AppScope: Application energy metering framework for Android smartphone using kernel activity monitoring,” in Proc. USENIX Conf. Annu. Tech. Conf., 2012, pp. 36–36.C.

4. P. Pearce, A. P. Felt, G. Nunez, and D. Wagner, “AdDroid: Privilege separation for applications and advertisers in Android,” in Proc. 7th ACM Symp. Inf. Comput. Commun. Secur., 2012, pp. 71–72

5. L. Ravindranath, J. Padhye, S. Agarwal, R. Mahajan, I. Obermiller,and S. Shayandeh, “AppInsight: Mobile app performance monitoring in the wild,” in Proc. 10th USENIX Conf. Operating Syst. Des. Implementation, 2012, pp. 107–120.

6. L. Batyuk, M. Herpich, S. A. Camtepe, K. Raddatz, A.-D. Schmidt,and S. Albayrak, “Using static analysis for automatic assessment and mitigation of unwanted and malicious activities within Android applications,” in Proc. 6th Int. Conf. Malicious Unwanted Softw., 2011, pp. 66– 72.

7. F. Ding, F. Xia, W. Zhang, X. Zhao, and C. Ma, “Monitoring energy consumption of smartphones,” in Proc. Int. Conf. Internet Things 4th Int. Conf. Cyber Phys. Social Comput., 2011, pp. 610–613.

8. A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, “Android permissions demystified,” in Proc. 18th ACM Conf. Comput. Commun. Secur., 2011, pp. 627–638.

9. A.Carroll and G.Heiser, "An analysis of power consumption in a smartphone," in Proc. USENIX Conf. USENIX Annu. Tech. Conf.,2010, pp. 21–21

10. K. Paul and T. K. Kundu, "Android on mobile devices: An energy perspective," in Proc. IEEE 10th Int. Conf. Comp. Inf. Technol., 2010, pp. 2421–2426

11. L. Zhang et al., "Accurate online power estimation and automatic battery behaviour based power model generation for smart-phones," in Proc. 8th IEEE/ACM/IFIP Int. Conf. Hardware/Software Codesign Syst. Synthesis, 2010, pp. 105–114.

12. "Contagio mobile, mobile malware mini dump." [Online]. Available: http://contagiominidump.blogspot.com

13. GoogleGroups, "Virustotal," 2015. [Online]. Available: https: //www.virustotal.com/

14. Dr.Web, "Android malware review," 2015. [Online]. Available http://news.drweb.com/show/review/?lng=en&i=9546

15. K. S. Labs, "Kindsight security labs malware report h1 2014," 2014. [Online]. Available: http://resources.alcatel-lucent.com/ ?cid=180437 15455971 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publi catio ns/rights/index.html for more information.