A Review Paper on SCANDROID and Its Challenges for Smartphone Devices

Miss. Kiran P. Jaiswal
Student
Department of Computer Science and Engineering
Hindi Seva Mandal's, Shri Sant Gadge Baba College of Engineering & Technology, Bhusawal - 425203, Dist. – Jalgaon, Maharashtra, India

Abstract— Android Smartphone devices have expanded an approval in current existences. Individuals stay affecting after PC/laptop (Supercomputer) to Smartphone devices. Nowadays Smartphone/ tablets remain essential part of our day-to-day lives later they afford us substantial quantity of facilities. the usage of Smartphone facilities improved expressively since of the connectivity delivered via Smartphone similar WiFi, GPRS, Bluetooth, 3G,4G etc. people are using Smartphone devices comparable computer's to store specific or trusted data, directing mail, web surfing, social networking, net banking etc. Consequently, Smartphone come to be indispensable attraction of enemies/hackers [2]. Leading enemies are attracted in the expensive private material. In Additional, Smartphone are continuously associated, which sorts them beneficial as bots in hats. Third, Smartphone can show superior Short message to be send (SMS) or SMS that contribute the victim to exorbitant facilities, and therefore straight generate currency for the enemy. It is awake to the Smartphone operating system (OS) to guarantee the safety of the data on the device [3]. an innovative methodology of improved security framework which can be exist cohesive with the remaining Android Security Framework to create Android more protected and to keep track of the files retrieved by one of the susceptible apps downloaded from altered foundations on the web. The suggested improved security framework increases the safety of Android File System by controlling the apps whose performance equals through the malware [4]. improving the wireless power allocation.

Key words: Scan droid, Operating system, Security, GPRS, Smartphone, Android app, Security Framework, File System, Malware, SMS, 3G, 4G, and PCC

I. INTRODUCTION

In this paper, we existent SCANDROID, is a device for Automated security authorization of Android solicitations. SCANDROID statically examines statistics movements over Android applications, and can make security related judgments automatically, based on such streams. In precise, it can resolve whether it is damage lesson behalf of an application to track with definite authorizations, built on the authorizations prescribed by further applications. On the other hand, it can afford sufficient framework to the user to make up-to-date security-relevant resolutions. SCANDROID can further suitable in several proof carrying code (PCC) settings. For example, applications can be studied offline through SCANDROID in an application store, and Android devices can pattern certificates of security delivered by the application store at install period. Otherwise, the developer can paradigm a security proof for the application by consuming our investigation, and the device can validate that proof before installing the application [1].

Fig. 1: Android Security

Android is a vulnerable source operating system namely source code of android is accessible to all. Consequently, it permits adjustment at kernel level. There is numerous antivirus software, intrusion recognition system, malware recognition, firewalls accessible as an application for android devices as shown in the figure 1. Certain security boosted Linux are implemented to improve android security. [2] Our discoveries designate that the Android authorization system is neither an overall success nor a whole failure. Because of low consideration and comprehension charges, authorizations alone do not guard most users from unwanted applications (i.e., malware or gray ware). Conversely, a sectional of laboratory training contributors (20%) confirmed Responsiveness of permissions and realistic rates of accepting (comprehension grades of 70% or higher) [5]. Google does not breakdown or hamper Android applications. As an alternative, android usages authorization to aware users to privacy- or safety offensive applications. When a user recruits the procedure of installing an application, he or she is displayed the list of agreements that the application demands. This list recognizes altogether of the phone properties that the application will require access to if it is installed. [6]

II. LITERATURE SURVEY

W. Enck, D. Octeau, P. McDaniel and S. Chaudhuri present 'a study of Android application security'. They introduce the decode compiler, which generate android application source code directly from its installation image. They design and execute a horizontal study of Smartphone applications based on static analysis of 21 million lines of recovered code. Their analysis uncovered pervasive use / misuse of personal / phone identifiers, and deep penetration of advertizing and analytics networks.

S. Powar, Dr. B. B. Meshram, surveyed on 'Android security framework', in this paper, they described android security framework. Increased exposure of open source Smartphone is increasing the security risk. Android provide a basic set of permissions to secure phone. The technique to

make Android security mechanism more versatile, the current security mechanism is too rigid. User has only two options at the time of application installation first allow all requested permissions and second deny requested permissions leads to stop installation6. S. Kaur and M. Kaur presented review paper on 'implementing security on Android application'. In that paper, they described how security can be improved in android based system so that users can safely use the android smart phones.

P. Gilbert, W. Enck, L.P. Cox, B.G. Chun, J. Jung, A.N. Shetland P. McDaniel presented 'TaintDroid: An Information-Flow Tracking System for Real-time Privacy Monitoring on smart phones'. Now days smartphone operating systems often fail to provide users with adequate control over and visibility into how third-party applications use their private data. They address these shortcomings with Taint Droid, system-wide dynamic taint tracking and analysis system capable of at the same time tracking multiple sources of private data. Taint Droid display real-time analysis by leveraging Android's virtualized Execution environment and Monitoring private data to inform use of third-party applications for phone users and valuable input for Smartphone security service firms seeking to identify Misbehaving applications.

B. J. Berger, M. Bunke, and K. Sohr presented an android security case study with Bauhaus. In this paper, they discovered that firms and corporation now uses security software for code analysis to discover security problems in application. They carried out a case study on android based mobile in cooperation with a security expert and employed the reverse engineering tool-suite Bauhaus for security assessment. During the investigation they found some inconsistencies in the implementation of the Android security concepts. Based on the case study, they propose several research topics in the area of Reverse engineering that would support a security analyst during security assessments [5].

In, MOSES is the first solution to provide policy-based security containers implemented completely via soft-ware. By acting at the system level we prevent applications to be able to bypass our isolation. How-ever, at the present moment MOSES has also some limitations. At first, fine-grained policies and allowed applications are specified using the UID of an application. Meanwhile, in Android it is possible that some applications share the same UID. Thus, if we apply MOSES rules and restrictions to one application they automatically will be extended to the other ones with same UID. Furthermore, some fine-grained policies in MOSES are built on top of Taint droid functionality. Thus, MOSES inherits the limitations of Taint droid explained in Section 3. It should be also mentioned that the applications that have root access to the system can bypass MOSES protection. Thus, MOSES is ineffective in combating with the malware that obtains root access, e.g., root kits [3].

III. SECURITY FRAMEWORK OF SMARTPHONE DEVICE

A Smartphone device is a complicated arrangement of a mobile phone and a computing platform, using high speed connectivity and dominant computing capability. Consequently, the Smartphone has required components of the computing platform: an operating system, applications, software and hardware. Moreover, as an individual Communication device, the Smartphone as well often takes multiple communication abilities and capability to accumulation great quantities of sensitive user data. Android is Linux established operating system related through software's. Linux kernel offers fundamental system facilities to android software organization or stack such as memory management, networking, process management, device drivers, and file system and power management. Software system encompasses android intuitive libraries which are printed in C/C++. Dalvik Virtual Machine is the core of android, is an essential libraries offer runtime environment for android. DVM runs. dex files which are compressed and well-organized. Android applications are wholly written in java and are wrapped in .apk documentation which grips completely code of app [2]. The consequential security framework is much further similar to a multi-user server than the sandbox initiate on the J2ME platforms. Contrasting a desktop computer operating system anywhere a user's applications all run under the equivalent UID (UNIX USER IDENTIFIER), Android applications are independently separated from all other. Android applications run in separate processes under different UIDs each with different set of authorizations. Programs require no approval to read or write each other's data or code, and sharing data/files among applications necessity be complete clearly by the programmer as shown in the figure 3. The Android GUI environment takes some unique and separate security structures that help maintenance this separation of procedures. Android framework encourages the user to agree a list of mandatory authorizations; the user might allowance all of the authorizations with the purpose of install the application effectively or reject the authorizations to withdraw the installation. Essentially, there are a number of security problems in such a framework:

1) The handle deceives and allowances all of the required permissions with the purpose of install the application effectively.
2) When the app is installed and permissions are approved; there is no appliance for limiting an application to withdraw the permissions previously approved.
3) Nearby is no approach of limiting access to the resources based on dynamic limitations as the authorization model is based on install-time checked first?
4) Approved authorizations can only be cancelled by uninstalling the application. At the period of installation, the user is available with a dialog box listing all authorizations demanded by the app to acquire successfully installed. These authorization demands are defined in an XML File called AndroidManifest.xml, which is transported through all Android app [4].

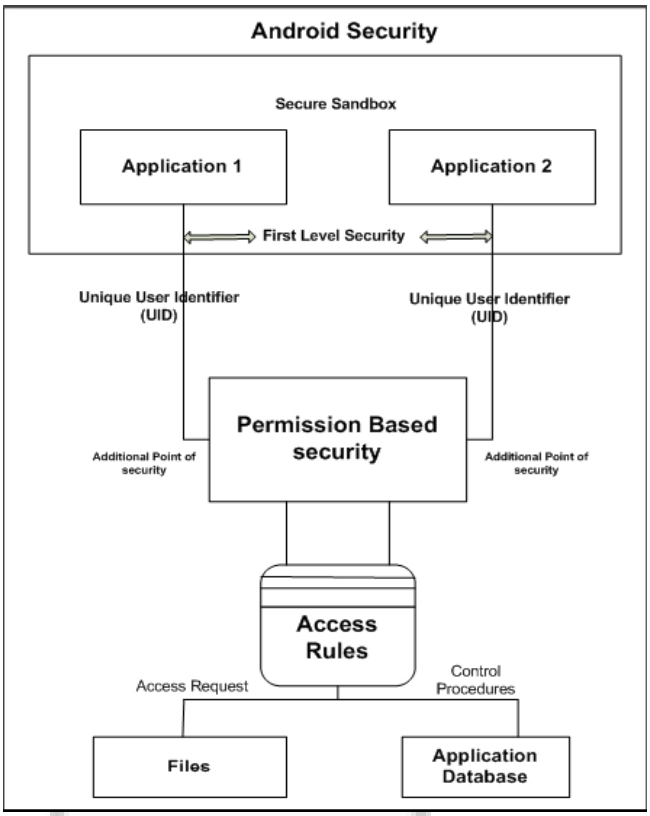

Fig. 3: Security Architecture of Android [7]

A. 1Methods to Attack on Smartphone Android Device:

1) Wireless Method:

Several categories of wireless attacks are there negotiating the thoughtful data of Smartphone device. The further most common attack is inquisitive on communication media to acquire the password like delicate information. Wireless attacks can exploitation the MAC address of device.

2) User based Method:

User must recognize the crucial of elementary security mechanisms. Attacker cans expenditure dissimilar behaviors to deception user's device. User must retain the trusted data in encrypted method. User should recognize mischievous links and reliable links to click and must know to preserve their password secret.

3) Worm based Method:

Smartphone device ensures about facilities with several connectivity outfits. Worm centered attack can be complete through Bluetooth, downloading diseased files, injecting diseased memory card, email, SMS/MMS etc. Well recognized way to extent the worms is Bluetooth connection. Worms can also attack on networks and give and take the phone to custom the network facilities.

4) Break in Method:

In this attack the attacker catch out the susceptibilities in kernels core libraries, applications and break in. attacker can increase full control of device. For example: Doom boot-this Trojan installs corrupted system libraries into C: drive of device.

5) Infrastructure based attack Method:

In this type of attack the attacker attacks on services like making/receiving calls or SMS. This type of attack can happen in GSM, GPRS, and EDGE networks.

6) Botnet Method:

Android Smartphone device can be portion of botnets. Attacker can practice Smartphone device as client to attack additional device or networks. [2].

IV. ANDROID SECURITY CHALLENGES IN SMARTPHONES

A. Android Platform Security:

Android runs on an extensive variety of devices and Android's security architecture depends on security structures that are embedded in the hardware. In Secure Boot The boot process of an Android device is a five-step process. First the CPU fright simple meting from its retune vector to which the initial boot-loader (IBL) code after the ROM is wired. Then the IBL loads the boot loader after the boot medium into the RAM and achieves a signature check to guarantee that only genuine code becomes executed. The boot loader loads the Linux kernel and as well executes a signature squared. The Linux kernel modifies all the hardware and lastly offspring the first user process called in it delivers a configuration file and boots the rest of the Android user land.

B. Android System Security:

The flash loading of an Android device is generally divided into multiple dividers. The system divider covers the Android base system such as libraries, the application runtime and the application framework. This partition is fixed read-only to avoid variation of it. This also permits a user to boot their device into a harmless mode which is unrestricted of third party software. Later Android 3.0 it is probable to encode the data partition with 128bit AES. To permit file system encryption the user takes to fix a device password which is used to expose the master key. Data Security by avoidance an application's files are private. They are owned by that application's separate UID. Obviously an application can create world readable/writable files which provide access to everyone. Applications after the same author can track with the same UID and thereby get access to mutual files. Files twisted on the SD card are world readable and writable. Later Android 4.0 the structure delivers a Keychain API which proposals applications the prospect to safely supply certificates and user identifications.

C. Android Security Enhancements with Android 4.2 and the resulting negligible statements Google presented different security landscapes in Android. The user nowadays can choose to confirm side-loaded applications earlier to installation. This is also identified as the on device Bouncer. It assessments for shared malware and warnings the user if the application is measured harmful. So remote the credit charges don't measure up with other profitable malware scanners [5].

ACKNOWLEDGEMENT

I would like to thank my honorable Principal Dr. R. P. Singh, my special thanks to my Seminar – II guide Prof. Y. S. Patil & Head of Department, Prof. D. D. Patil, and all the respected teaching faculties of Department of Computer Science & Engineering of Hindi Seva Mandal's, Shri Sant Gadge Baba College of Engineering & Technology, Bhusawal. Also I would like to thank my parents, friend for motivating me for this paper work activity. My special thanks to all the writers or authors of reference paper that are referred by me.

V. CONCLUSION

With the existing security architecture, several Smartphone operating systems are susceptible to attacks since the Smartphone user is involved in determining which applications will be installed on the phone device. It is not informal for a user to judge applications by their explanation. The Android framework is single platform that assumes the user to be security aware and implicitly accepts applications developers are not malicious [4]. The Incorporation of technologies into an application certification process wants disabling logistical and technical challenges. Android delivers additional security than other mobile phone Stages [5].

REFERENCES

[1] Adam P. Fuchs, Avik Chaudhuri, and Jeffrey S. Foster "Scan Droid: Automated Security Certification of Android Applications" 2015.
[2] Chetan C.Kotkar, Pravin "Exploring Security Mechanisms to Android Device" International Journal of Advanced Computer Research Volume-3 Number-4 Issue-13 December-2013.
[3] Mis.Prajakta S. Deshbhratar, Prof. Mayur S. Burange "Android Security –Big Challenge "International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-3, Issue-3).
[4] Muneer Ahmad Dar ,JavedParvez "A Novel Strategy to Enhance the Android Security Framework ".International Journal of Computer Applications (0975 – 8887) Volume 91 – No.8, April 2014.
[5] TiwariMohini, SrivastavaAshish Kumar and Gupta Nitesh "Review on Android and Smartphone Security" Research Journal of Computer and Information Technology Sciences ISSN 2320 – 6527 Vol. 1(6), 12-19, November (2013).
[6] Adrienne Porter Felt, Elizabeth Hay, Serge Egelman, Ariel Haneyy, Erika Chin, David Wagner "Android Permissions: User Attention, Comprehension, and Behavior"
[7] By Muneer Ahmad Dar &JavedParvez "Evaluating Smartphone Application Security: A Case Study on Android" Global Journal of Computer Science and Technology Network, Web &Security Volume 13 Issue 12 Version 1.0 Year 2013.
[8] Cassandra Beyer "Mobile Security: A Literature Review " International Journal of Computer Applications (0975 – 8887) Volume 97– No.8, July 2014.
[9] Mr. Sumedh P. Ingale1, Prof. Sunil R. Gupta "security in android based Smartphone" International Journal of Application or Innovation in Engineering & Management (IJAIEM) Volume 3, Issue 3, March 2014.
[10] Tushar Chopra ,Vikash Kumar Sharma" Droid Guard: An Approach to Make Android Secure " International Journal of Computer Applications (0975 – 8887) Volume 117 – No. 8, May 2015.
[11] Jagdish Prasad Achara, Claude Castelluccia" Smartphone security overview" INRIA Rhone-Alpes5 December 2012.

