An Enhancing to Upgrade Device Level Security Approach for Android via Keychain, Spitting Apps Techniques Using UAS

[1] Prof.Shraddha Mankar[2] Prof. Priyanka Kokare[3] Prof. Balaji Chaugule
[1]Assistant Professor[2]Assistant Professor[3]Assistant Professor Computer Department
ZE'S ZCOER, Narhe, Pune-411041

Abstract: In this widely spread new era, Smartphone modern Operating systems are Android & ios are one which use of Mobile platforms more increasing rapidly. In all over the world, we just discussed and focused on their part of loaded inherent security and privacy by Comparative study under application development with their individual Features, secure and Threats level models of OS to protect from vulnerability under by detecting malware attacks. In this paper, we signify and focused instead of interacting to how android and ios Smartphone Operating System are taken protection over the world, under criminals malware detection of malicious applications attack, results of their study showed high percentage of such attack in the Smartphone's device those are built on Operating System with many vulnerabilities are revealed? but we discussed on the behalf part of device manufacturer phase (write device driver) come face to new advance security challenges you need to address with integrated design/architecture and the problem faced with the ugly truth behind android upgrade problem under device-level, the security model describe user data with secure keychain services, apps spitting technique, reverse engineering etc. utmost Device (mobile phone) manufacture make their money by selling phones only right? And surprisingly, selling phones remains their primary focus instead make compatibility related to an Android platform of advance device-level security for any kind of operating system security.

Keywords: Android ecosystem mobile device management, optimization of mobile device packages, keychain services, App spitting techniques, UAS, reverse engineering.

1. Introduction

Android is famous operating system and developed as an open source operating system. Open source it means owner/developer can change and adapt the source code data of their device like android be the owner of many devices.[1] user wants flexibility to change the way of running process for their Devices apparently in parallel process they can vulnerable to attacks while altering their source code of devices[1].user accidently belonging to cybercrime insured unlikely, we take just review of Ios operating system with running process on their own Apple-branded manufactured products mobile device with certified proof mining store by host companies. Manufacturer plays a most important role while providing a hardware device to keep more secure than others belongs to os[1][2]. Moreover as the manufacturer should be use ritual space which customized ROM or parent/base Operating system that has the software installed that could not effortlessly be concerned or analyzed for malicious error indented to do. Just overlook as Android OS is hugely popular, means developer upgraded constantly new apps designed to run on any platform. Chaos makes when hackers create app design to infect your mobile device because device not that much compatible to user authentication service (UAS) based on security purpose there should be an app review process for Google play. As we said," android open play store is market place is the biggest blessing and darkest curse "also claim to have significant stride currently in the efforts to curb the amount of malware on the business card. In addition, the Android user modifies their device setting to allow apps for acceptance from outside the app store but such a way creates the greatest risk of malware attacks [4]. Unfortunately, by stringent process than what manufacturer faced while involving the addition or an accident installed an app for a malicious sneak on to the Google play store likewise enable to install software from unknown sources[3]. We stated as disadvantage of apple is less flexible because of it doesn't allow the owners to modify or upgrade source code of its ios operating system or custom ROM to encumbered on their own device space that makes system more secure and protective via scheming the complete experience what they faced .providing full support to accept such device is less tempting target to hackers and cybercriminals could make certain victim if they focused on attack malware on own devices.

If you value mobile device security of Android, probably noticed or want to compare android Vs ios .how they measure up across the range of different features like apps marketplace security when it comes to mobile manufacturer devices. More user more target and for hackers more reason to developed malware. The smaller number of target heighten security, make somehow less attracted by hackers and device keep more secured so, through this paper really need to promote device security authority acceptance in the wider range.[5][6]

II. Related Research Work

In some circles it can be found by the researchers, Platform based Operating system of Apple's iOS has extensively measured the more security of the closed operating systems because it doesn't unconfined or released its default source code for apps developers/owners of iPhone system which can't modify the source code of their phones model themselves. To find more vulnerabilities hackers get more difficult for ios power devices rather than Android devices which are completely opposite relying on open source code might be created weakness in their device-level security. possibility to hit the cybercriminals unleashes and be careful while downloading an app from third-party stores rather trusted store such as Google play, apple apps store, yet the apps they sell.[1]

If we think by social engineering task there are more cybercriminals attempt and many more trick created for users such as giving up personal login information, access bank account details, and other private data which doesn't matter what operating system you are using rather it completely depend on mobile device manufacturer with their UAS for compatible and advance device for enhance their level of security.

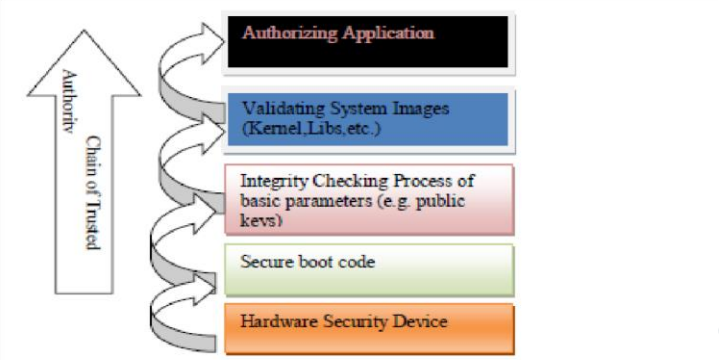

Figure 1.Architecture of secure boot

There is further need to keep your mobile device/phone or tablet safe n secured by vulnerable attacks we should always upgrade our mobile device to a latest released version of their operating system but it also completely depends on the device compatibility to upgrade OS. Regular release updated device should modify with their current version by default not manually because in busy schedule tempting to skip or avoid up gradation which released by owner and make the biggest mistake to allow or involve malware against latest security threats[2][3]. ignoring updates could put the device on risk. In all off course user should turn on automatic updates of their own devices to released automatically latest version of their operating system by advance manufactured device task.

III. Literature Survey

As we said earlier device manufacturer play an important role to keep more secure and protect from a vulnerable attack such as malware detection process. hardware integration be the central or core part for make sure built-in security features on device work properly or not as much case of Samsung which has Knox refuge solution comes to preinstalled in among all of the company phone, tablets, wearable devices etc. such a kind of platform provide as much as secure booting process, helping to avert unauthorized or illegal access to software allowance from loading and downloading when user turns on or restart the Samsung mobile phone device via UAS system[3].

To keep more security and safety we should more cautious about protection never attract towards operating system app stores with their features while downloading apparently rogue app take simple way to infect device like android tablets device especially downloading from third-party app store source which is no reputed or certified. So, we should keep avoiding or ignoring downloading from outside Google store to increase infected downloading and keep the device on a higher risk [7].

To ensure OS devices and providence of security features implemented from hardware level to software stack which is shown by given following diagram as, Security architecture Diagram for hardware device:

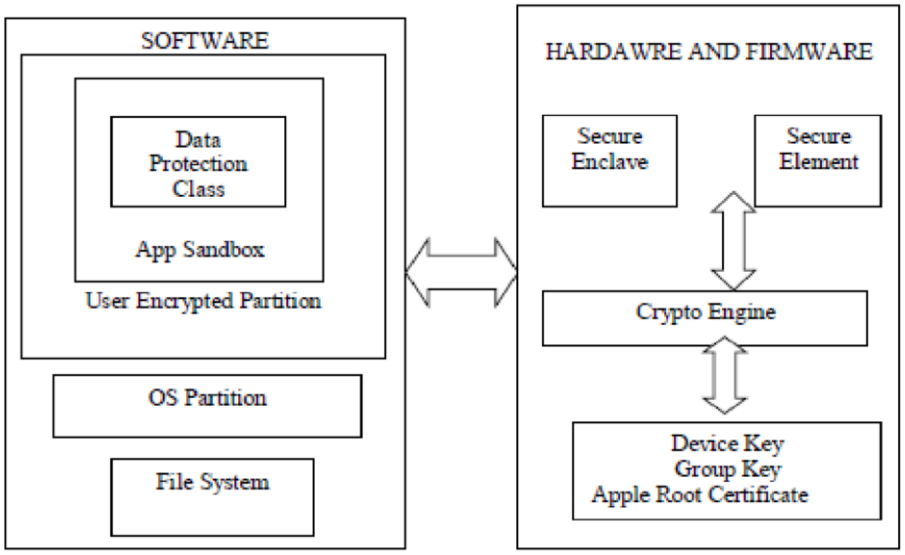

Figure 2: To Provide Visual Overview of the Different Technology through Secure Devices by Security Architecture Diagram

At the moment, companies are desperately trying to collect the data which should put appropriate sensitive process being transmitted by private setting default while a user might not know the way in all particulars, notable privacy concern. As above maintained architectural diagram shows hardware and software interaction through spitting up the security model into many layers as roughly which can describe as follows;[7][4].

First and most is Device-level-security which shown the hardcore part of the malicious attack by a hacker. We need to take care and tends to form secure enclave as co-processor fabricated within System on chip (SoC) which utilized to provide authentication also has its own secured boot layer firmly separated by application processor which can be done highly encapsulated task including key management, processing cryptography operations with top maintain high data integrity. Once we start up with process containing components which cryptographically signed by an authority trusted chain and verify processer authority. Each secure enclave with a unique id (UID) during the manufacturing process.UID used to generate and keep secure enclave memory space protected and data of their files stored in the file system is encapsulated also responsible for decrypting, encrypting system file contains all keys & processing done as below we describe in file encryptingkeyhierarchy[3].

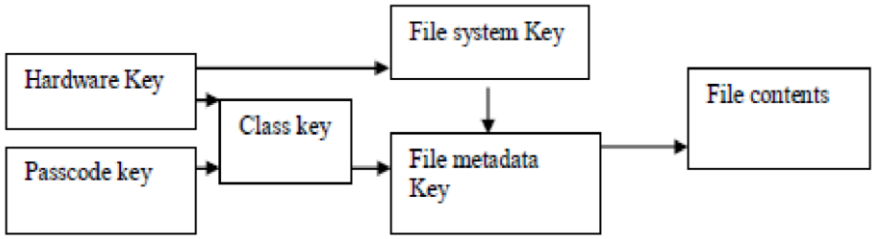

Figure 3.File Encryption Key Hierarchy

As hardware security features CPU consumption and cryptography operation becomes complicated also they could introduce battery life problem. With the biggest issue reveals device-level security part to ensure and make encryption process highly efficient Unique UID and Device group ID (GID) created common processor tied to a particular device via protecting file data by construct and manage a hierarchy of key in conjunction with hardware encryption engine with related part of hardware security conception[7][2].

Once we create a file, the data protection Device system creates 256-bit keys forward to hardware Engine i.e. AES, which will use the particular file to encrypt separate data and Protect/secure a device to make sure constraint of it.

IV. Proposed System A. Overview of Keychain and app spitting:

The security model ensures at device level unauthorized personnel with securing user data with formally generated keychain concept with the use of persistent data in apps using their services, also can able to structure Keychain API device which can illustrate easy and quick way to start functionalities in your app in implementation of a wrapper .keychain offers secure alternative to save very sensitive data, such as user id and password methods provided by keychain services to deal with this problem developer safely handle sensitive information.

Keychain is a great concept to protect device because the part of data encryption does automatically take care of it before it the place to store in the file system so, there is no need to much waste extra time for building and upload encryption algorithms. Keychain is configured to lock system which is impossible to access and decrypt store keychain items. The device is locking while keychain lock, device unlock even when it is unlocked, once they configured it [4].We can express app spitting technique that divides and make partition into the number of collaborating apps to expose information flow of os level mechanism to allow security policy separately with implementing too as AAPSAW to show the method to form an automatic selecting code of partition that isolates the permission[9][10].

B. UAS (User approved Security) System:

The main Objective is to provide novice user to take permission with more be in charge of over control at the moment in time of implementation in order to enhance the security level of the device.UAS itself extract a various file from APK file using reverse engineering process as an eg.androidmenifest.xml list all permission given by system user will accept or reject the access of resource or from the resource code.

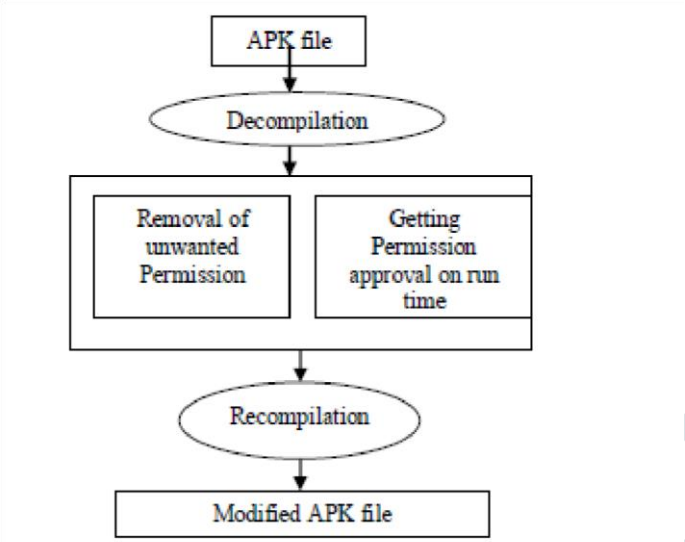

Figure 4: Structure Of UAS System

C. Reverse Engineering:

We need to suggest and put to extract file systems using the part reverse engineering concept by compiling each Android app package in file system under single file .apk (Android application package) include all sources code & resource file called .dex file in order to take out or extract various file from .apk file downloader and from the web source file system data as;

classes_dex2jar.src jd-gui-0.3.5.windows SignApk apktool1.4.1.tar apktool-install

For e.g. FileMuneer.apk using the above commands: “apktool if Muneer.apk” and then type “apktool d Muneer apk path output”.

Once we extract a variety of file from .apk file store our system will allow to remove unwanted and give extra permission from androidmenifest.xml then we have to recompile repackage apps to create the modified .apk file with user authentication services in that place .once we use the net.x comment for recompilation to make a built-in process to create the new folder in the decompiled path. Generate the new .apk file will be inherent in on the same path still personalized app would not work appropriately & try to run it into a system. Then, ROM will hold up in the BOOT process system. In order to come away from this, we necessitate using signapk.jar and type “java –jar signapk.jar certificate.pm key pk8 our_repackaged_new_apk”[8]. in all for making sure process we need to recognize the source code to protect device via secure data code.

V. Future Scope

The Most Common Complaint About Android Device Is Slow Up Gradation Which Big, Hot Mess. So, In Future We Need To Used Advance Key chains, App Splitting Technique With User Authorized System. Over The World Of Google Itself, Only Manufacturer Focused On Satisfactory Compatibility Grade For Its Performance And Level Of Device Building Task. Regarding Company Selling Phone Have No Real Motivation To Care About Post Sale Support.MDM Mobile Device Management Course Should Me Certified By Manufacturer To Learn How To Secure Mobile Device Without Degrading User Experience. A Company Like Nokia, Samsung Make Compatible Upgraded Hardware Device. What Company Offered In Terms Of Quality & Experience Was Almost Irrelevant. So, In All, We Should Focus On Company Manufacturer Device to Secure Providence

VI. Conclusion

In this paper we need to focused on mobile device management for advance devicelevel security purpose using UAS user authentication service through Android device manufacturer phase(write device driver) by collecting information on purpose of parented Common Criteria Certification with commercial solution for classified (CSfC), Cryptography validation with ISO Certification etc. comes along with face to new security challenges you need to address with integrated design/architecture and the problem faced with the ugly truth behind android upgrade problem under device level, the security model describe user data with secure keychain services, apps spitting techniques, reverse engineering etc. utmost Device (mobile phone) manufacture make their money by selling phones only right? And surprisingly, selling phones remains their primary focus instead make compatibility related to android platform operating system security.

VII. REFERENCE

[1] iOS Security Guide of iOS 12.1 in November 2018, Apple Books, HealthKit, HomePod, Sirikit, TrueDepth, and tvOS are trademarks of Apple Inc.

[2] November 2018 [1] K. Jamdaade1, A.Khairmode, and S. Kamble, "A Comparative study between Android & iOS" in International Journal of Current Trends in Engineering & Research (IJCTER) e-ISSN 2455–1392 vol 2, no. 6, pp. 495 – 501, June 2016

[3] I.Mohamed and D. Patel, "Android vs iOS Security: A [2] Comparative Study", in IEEE 12th International Conference on Information Technology - New Generations (ITNG), INSPEC Accession Number: 15180414, DOI: 10.1109/ITNG.2015.12, 2015

[4] Mohd Shahdi Ahmad "Comparison Between Android and iOS Operating System in terms of Security" - 2013 8th International Conference on Information Technology in Asia (CITA)

[5] Fattoh AI-Qershi "Android vs. iOS: The Security Battle" - 2014 IEEE

[6] S. Annapurna, K.V.S. Pavan Teja, Y. S. Murty, "A Comparative Study on Mobile Platforms (Android vs. IOS)", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 5, no. 3, March 2016

[7] Google. Smartphone Operating System: http://www.android.com!

[8] https://www.apple.com/privacy/docs/iOS_Security_Guide_Oct_2014.pdf

[9] AppleCare, App Store, iCloud, iCloud Drive, iCloud Keychain, and iTunes Store are service marks of Apple Inc., registered in the U.S. and other countries.

[10] Enhancing Android Security Through App Splitting Drew Davidson1(B), Vaibhav Rastogi2, Mihai Christodorescu3, and Somesh Jha1,2 .1 Tala Security, 200 Brown Road, Fremont, CA 94539, USA drew@talasecurity.io.