A Survey of Android Technology

Shivam[1], Ranjana sharma[2]

[1]B.sc(H) 6thsemester,CCSIT,TMU,MORADABAD
[2]Assistant Professor, CCSIT, TMU MORADABAD
[1]shivamarora2015@gmail.com
[2]sharmaranjana04@gmail.com

ABSTRACT: **Android is a mobile operating system (OS).Currently developed by Google, based on the Linux kernel and designed primarily for touch screen mobile device such as smart phone and tablet. Android is a software stack for mobile device that includes an operating system, middleware and key application. Android is a software platform and operating system for mobile device based on Linux operating system and developed by Google and the Open Handset Alliance.**

Before Android, mobile developers faced many roadblocks when it came to writing applications. Building the better application, the unique application, the competing application, the hybrid application, and incorporating many common tasks such as messaging and calling in a familiar way were often unrealistic goals. It allows developers to write managed code in a java like languages that utilize Google developed Java libraries, but does not support programs developed in native code. The unveiling of the Android platform on 5 November 2007 was announced with the founding of the Open Handset Alliance a consortium of 34 hardware, software and telecom companies devoted to advancing open standard for mobile device .When released in 2008,most of the Android platform will be made available under the Apache free software and open source licence

Keyword: Linux kernel, Open source licence, Native code, Stack, Open Handset Allince

1. INTRODUCTION

Android is a modern mobile platform that is designed to be truly open source. Android applications can use advanced level of hardware and software, as well as local and server data, exposed through the platform to bring innovation and value to consumers. Android platform must have security mechanism to ensure security of user data, information, application and network.

Open source platform needs strong and rigorous security architecture to provide security. Android is designed with multi-layered security that provides flexibleness needed for an open platform, whereas providing protection for all users of the platform designed to a software stack, android includes an operating system, middleware and core application as a complete. Android powers hundreds of millions of mobile devices in more than 190 countries around the world.

Android architecture is designed with keep ease of development ability for developers. Security controls have designed to minimize the load on developers. Developers have to simplywork on versatile security controls. Developers are not familiar with securities that apply by defaults on application.

Android is also designed with focused on user's perspective. Users can view how applications work, and manage those applications.

ANDROID PLATFORM SECURITY ARCHITECTURE

Android seeks to be the most secure and usable operating system for mobiles by re-purposing classical operating system security controls to protect user data, system resources and provide application isolation.

Android provides following security features to achieve these objectives are first robust security at the operating system level through the Linux kernel, second compulsory application sandbox for all applications, third secure interposes communication, fourth application signing, and sixth application defined permission and user have to grant permissions.

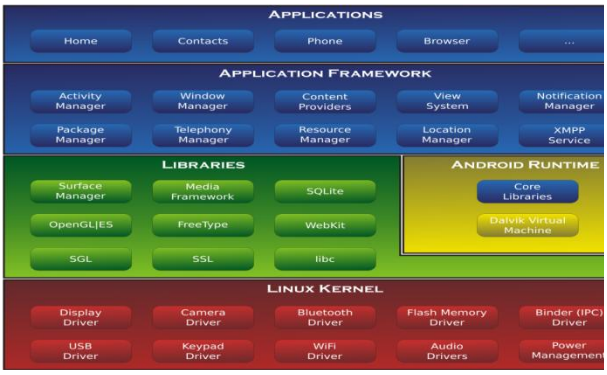

Fig –Android Architecture

Figure summarizes security components and considerations at the various levels of the Android. Every component assumes that component below is properly secured. With exception some Android operating system code running as root, all process run above the Linux Kernel is restricted by the Application Sandbox.

SECURITY IN ANDROID:

i. Android is open source platform, developers will work along to enhance it1.

ii. Android platform is multitasking software; therefore no application will gain critical access to the components of OS3.

iii. Android platform is UNIX based operating system that is the most secure operating system1.

iv. The developers need a unique signature to publish their application on market4.

v. Users will report a possible security flaw through their Google account.

vi. All applications on android need permission from the user at the time of installation.

SECURITY ISSUES FACED BY ANDROID

Android is not secure as it appear, even when such robust security measures. There are several security problems faced by the android, some of them are mentioned below.

i. Android has no security scan over the apps being uploaded on its market.

ii. There are some apps which can exploit the services of another app without permission request.

iii. Android's permission security model provides power to user to make a decision whether an app should be trusted or not. This human power introduces a lot of risk in Android system.

iv. The Open Source is available to legitimate developers as well as hackers too. Thus the Android framework cannot be trusted when it comes to develop critical systems.

v. The Android operating system developers clearly state that they are not responsible for the security of external storage.

Any app on the android platform will access device data just like the GSM and SIM marketer Ids while not the permission of the user.

Android platform provides all security features, but there will always be a risk if the user will install suspicious apps or allow permission to an app without paying attention.

LITERATURE SURVEY

W. Enck, D. Octeau, P. McDaniel and S. Chaudhuri present 'a study of Android application security'. They introduce the ded decompiler, 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 ofpermissions 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 installation.

S. Smalley and R. Craig presented 'Security Enhanced (SE) Android: Bringing Flexible MAC to Android'. The android software stack for mobile devices defines and enforces its own security model for apps through its application-layer permissions model. However, at its foundation, android depends upon the UNIX operating system kernel to shield the system from malicious or imperfect apps and to isolate apps from each other. At present, android leverages UNIX operating system discretionary access control (DAC) to enforce these guarantees, despite the notable shortcomings of DAC. In this paper, they motivate and describe their work to bring flexible mandatory access control (MAC) to Android by enabling the effective use of Security Enhanced Linux (SELinux) for kernel-level MAC and by developing a set of middleware MAC extensions to the Android permissions model.

P. Gilbert, W. Enck, L.P. Cox, B.G. Chun, J. Jung, A.N. Sheth and P. McDaniel presented 'TaintDroid: An Information-Flow Tracking System for Real-time Privacy Monitoring on smartphones'. 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 TaintDroid, system-wide dynamic taint tracking and analysis system capable of at the same time tracking multiple sources of private data. TaintDroid 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.

RESEARCH FINDING

Android has two basic methods of security enforcement. Firstly, applications run as Linux processes with their own user IDs and thus are separated from each other. This way, vulnerability in one application does not affect other applications. Since Android provides IPC mechanisms, which need to be secured, a second enforcement mechanism comes into play. Android implements a reference monitor to mediate access to application components based on permission. If an application tries to access another component, the end user must grant the appropriate permissions at installation time.

Phone identifiers are leaked through plaintext requests. Phone identifiers used as device fingerprints. Phone identifiers, specifically the IMEI, are used to track individual users. The IMEI is tied to personally identifiable information (PII). Not all phone identifier use leads to ex-filtration. Phone identifiers are sent to advertisement and analytics servers.

Using state-of-the-art tools for finding security bugs cannot reveal logical security problems such as undesirable interactions between components. With increasing complexity of software, software companies need to understand the security risks oftheir code, and tools employing program comprehension functionality will support them with this challenging task.

A study of android application security finding of exposure of phone identifiers and location are consistent with previous studies; analysis framework allows observing not only the existence of dangerous functionality, however conjointly how it occurs inside the context of the application. However, the integration of those technologies into an application certification process needs overcoming logistical and technical challenges.

Enhancing security of Linux-based android devices, Open source APIs of android may result in benign and malicious research activities hopefully resulting in an excellent safer smartphone platform.

L4Android: a generic operating system framework for secure smartphones: In this title they presented a generic OS framework that facilitates the creation of secure smartphone systems. The framework consists of three core components. A microkernel acts as the secure foundation and is accompanied by a user mode runtime environment. The third component is VMs to securely encapsulate existing smartphone operating system.

They implemented the core components of their framework on a mobile x86 and ARM platform. They evaluated framework by showing how it can be applied to available as the open source L4 solve four challenges in smartphone security such as secure software smartcards, and a unified corporate and private mobile phone.

Researches identified two fundamental causes of the attacks in WebView: weakening of the TCB and sandbox. They have shown that the condition for launching attacks is already matured, and the potential victims are in the millions. In their on-going work, they are developing solutions to secure WebView.

Android users need a way to determine if applications are leaking their personal information. They created a mapping between API calls and the permissions they must have to execute. AndroidLeaks is capable of analyzing 24,350 in 30 hours. AndroidLeaks drastically reduces the number of applications and the number of traces that a security auditor has to verify manually.

Android open source software and programmable framework behavior make it vulnerable to virus attacks20. The title takes into consideration the fact that Smart phones are memory, battery and speed constrained and hence exploiting the cloud to do the reputation index computation of a given application. By referring to the calculated matrix of reputation built by a given application, the model will notify users on the risk of the application before installation. Applications can be classified as highly risky, medium risk, less risk and genuine all based onreputation they have built in the cloud. The experimental results show that some application need to be regarded as highly risky and therefore warn users not to install them until they improve their reputation by passing the threshold set by the reputation based security model.

An android application sandbox system for suspicious software detection: In this title they presented a sandbox created for analyzing Android applications applicable as cloud service. Unlike other sandboxes, they added a pre-check technique that can analyze Android executables in a fixed manner. This can reports usage of malicious patterns within source code. The dynamic analysis can logged system calls from application. These can be used for further detections, either performed manually or automatically.

User can express and enforce fine-grained policies with temporal, spatial, and cardinal conditions that refer to both single representations of data and, via taint tags, to all representations of a data item. There system helps defend against two attacker models: malicious apps and malicious users. Their system considers the information flow in intents and content provider requests, because this alone is not sufficient, additional hooks were placed to observe the information flow between apps and the file system, the network and remote services (IPC). The Security-Manager is deployed as an integral part of Android and cannot be uninstalled by the user. Authentication between the Security-Manager and the monitor is achieved by Android’s IPC mechanism Binder.

They evaluated security and performance of their system. The security evaluation showed that the system can be considered as secure, in a sense that it is not possible for attackers to circumvent the monitor under the stated assumptions. The performance overhead was shown to be in an acceptable range for realistic end-user scenarios.

Android devices are complex, vulnerable, and attractive targets for attackers because of their broad application domain. The need for strong protection is apparent, preferably using multiple and diverse attack detection measures. Their security model performs attack detection on remote servers in the cloud where the execution of the software on the phone is mirrored in a virtual machine.

The evaluation of a user space implementation of our architecture Paranoid Android, shows that transmission overhead can be kept well below 2.5KiBps even during periods of high activity (browsing, audio playback), and to virtually nothing during idle periods. Battery life is reduced by about 30%, but they show that it can be significantly improved by implementing the tracer within the kernel. They conclude that our architecture is suitable for protection of mobile phones. Moreover, it offers more comprehensive security than possible with other models.

There is danger of malware for smart phones. Publicly available APIs can lead to new malwares that are able to extract various private data as well as to perform harmful action on infected devices. Private information is the number one data on mobile phones, and hence, a loss or modification will harm every affected person. But, as less and less critical malwares appear, security consideration seems to lose their importance. This is a big mistake and underestimating smartphone malware can cause serious problems not only concerning privacy issues.

CONCLUSION

Now days more than 1 million Android device activated. Android has very few restrictions for developer, increases the security risk for end users. In this paper we have reviewed security issues in the Android based Smartphone. The integration of technologies into an application certification process requires overcoming logistical and technical challenges. Android provides more security than other mobile phone platforms. Kirin will help mold Android into the secure operating system needed for next-generation computing platforms.

REFERENCES

[1] Android Open Source Project. Android Security Overview. http://source.android.com/devices/tech/security/index.html. (2013)

[2] Kaur S. and Kaur M., Review Paper on Implementing Security on Android Application, Journal of Environmental Sciences, Computer Science and Engineering & Technology, 2(3), (2013)

[3] Android Open Source Project. Security and permissions.

http://developer.android.com/guide/topics/security/permissi ons.html. (2013)

[4] Android Open Source Project. Publishing on GooglePlay. http://developer.android.com/distribute/googleplay/publish/ preparing.html. (2013)

[5] Enck W., Octeau D., McDaniel P. and Chaudhuri S., A Study of Android Application Security, The 20th USENIX conference on Security, 21-21, (2011)

[6] Powar S., Meshram B. B., Survey on Android Security Framework, International Journal of Engineering Research and Applications, 3(2), (2013)

[7] Smalley S. and Craig R., Security Enhanced (SE) Android: Bringing Flexible MAC to Android, www.internetsociety.org/sites/default/files/02_4.pdf . (2012)

[8] Enck W., Gilbert P., Chun B.G., Cox L.P., Jung J., McDaniel P. and Sheth A.N., TaintDroid: An Information- Flow Tracking System for Realtime Privacy Monitoring on Smartphones, 9th USENIX Symposium on Operating Systems Design and Implementation. (2010)

[9] Berger B.J., Bunke M., and Sohr K., An Android Security Case Study with Bauhaus, Working Conference on Reverse Engineering, 179–183 (2011)

[10] Ongtang M., McLaughlin S., Enck W. and McDaniel P., Semantically Rich Application-Centric Security in Android,Computer Security Applications Conference, 340–349 (2009)

[11] Schmidt A.D., Schmidt H.G., Clausen J., Camtepe A., Albayrak S. and Yuksel K. Ali and Kiraz O., Enhancing Security of Linux-based Android Devices, http://www.dai-labor.de/fileadmin/files/publications/lk2008-android_security. pdf (2008)

[12] Marforio C., Francillon A. and Capkun S., Application Collusion Attack on the Permission-Based Security Model and its Implications for Modern Smartphone Systems, ftp://ftp.inf.ethz.ch/doc/tech-reports/7xx/724.pdf (2013)

[13] Lackorzynski A., Lange M., Warg A., Liebergeld S., Peter M., L4Android: A Generic Operating System Framework for Secure Smartphones, 18th ACM Conference on Computer and Communications Security, 39-50 (2011)

[14] Luo T., Hao H., Du W., Wang Y. and Yin H., Attacks on WebView in the Android System, 27th Annual Computer Security Applications Conference, 343-352 (2011)

[15] Barrera D., Güne H., Kayacık S., Oorschot P.C. van and Somayaji A., A Methodology for Empirical Analysis of Permission-Based Security Models and its Application to Android, 17th ACM conference on Computer and communications security, 73–84 (2010)

[16] Gibler C., Crussell J., Erickson J. and Chen H., Android Leaks: Automatically Detecting Potential Privacy Leaks In Android Applications on a Large Scale, 5th international conference on Trust and Trustworthy Computing, 291-307 (2012)

[17] Burguera I., Zurutuza U. and Tehrani S.N., Crowdroid: behaviour-based malware detection system for Android, 1st ACM workshop on Security and privacy in smart phones and mobile devices, 15-26 (2011)

[18] Polla M.L., Martinelli F., and Sgandurra D., A Survey on Security for Mobile Devices, Communications Surveys & Tutorials, IEEE, 15(1), 446–471 (2013)

[19] Tesfay W.B., Booth T., and Andersson K., Reputation Based Security Model for Android Applications, Trust, Security and Privacy in Computing and Communications, IEEE Computer Society, 896-901 (2012)

[20] Bing H., Analysis and Research of Systems Security Based on Android, Intelligent Computation Technology and Automation, 581–584 (2012)

[21] Android Open Source Project. What is Android? http://developer.android.com/about/index.html (2013)