A Review on Android Operating System with its Security Features

Nilesh N. Chawande

M.E - (Computer science and Engineering)

Amravati, Maharashtra, India
NileshChawande@gmail.com

---***---

ABSTRACT : *Android package is one among the foremost wide used package of late. humanoid package is largely AN package for mobiles and is chop-chop gaining market share, with dozens of sensible phones and tablets either discharged or set to be discharged. it's mobile package that uses a changed version of the UNIX kernel a pair of.6. Google developed humanoid as a part of the Open phone Alliance, a bunch of over thirty mobile and technology corporations operating to open up the mobile phone surroundings. Android's development kit supports several of the quality packages utilized by seawall, thanks to that reality and Jetty's modularity and light-weight foot print it was potential to port seawall to that in order that it'll be ready to run on the humanoid platform. Humanoid package is principally divided into four main layers: the kernel, libraries, application framework and applications. Its kernel is predicated on UNIX. UNIX kernel is employed to manage core system services like computer storage, networking, drivers, and power management. In these paper totally different options of design of humanoid OS in addition security measures of humanoid OS square measure mentioned.*

Keywords **–** Android, version history, android security, SSL, features, service, Dalvik VM, Linux, Sandbox.

1. INTRODUCTION

Android package is one among the foremost wide used mobile package currently [1]. Automaton mobile package is predicated on the UNIX operating system kernel and is developed by Google. Automaton package is primarily designed for smart phones and tablets. Since automaton is associate degree open supply it's become the quickest growing mobile package. Thanks to its open nature it's become favourite for several customers and developers. Further- more package developers will simply modify and add increased feature in it to satisfy the most recent necessities of the mobile technology [2]. Automaton users transfer quite one.5 billion applications and games from Google Play monthly. Thanks to Its Powerful development framework users yet package developer area unit able to produce their own applications for big selection of devices [3]. a number of the key options of automaton package are: Application Frame work, Dalvik virtual machine, Integrated browser, Optimized Graphics, SQLite, Media Support, GSM Technology, Bluetooth, Edge, 3G, Wi-Fi, Camera and GPS etc [1]. to assist the developers for higher package development automaton provides automaton package development kit (SDK). It provides Java programming language for application development [1]. The automaton package development kit includes a computer programme, libraries, a French telephone individual supported QEMU (Quick Emulator), documentation, sample code, and tutorials [4]. Android could be a freely downloadable open supply package stack for mobile devices that has associate degree package, middleware and key application supported UNIX operating system and Java. Google purchased the developer of automaton in 2005, and automaton was unveiled in 2007. Google discharged the automaton code as ASCII text file below the Apache License. Automaton has varied developers writing applications (apps) everywhere the planet. Initial of all the developers write their script in Java, and so transfer the apps from the third party sites or on-line stores. In Feb 2012, 450,000 apps were on the market for automaton however the calculable variety of downloads since Dec, 2011 was quite ten billion. There area unit over three hundred million Androids in use and over 850,000 devices activated a day. Automaton is that the one among the foremost used mobile package with a market share of forty eighth and Over 400,000 applications on the market in Google play store. Automaton apps are put in over ten billion times and canopy a massive vary of classes from games and diversion to money and business services. Automaton package development and therefore the Google Play Market are comparatively open and unrestricted. This offers each developers and users a lot of flexibility and freedom, however conjointly creates vital security challenges.

2. VERSION HISTORY

Android is change day by day since its unleash. These updates to the bottom package primarily specializing in fixing bugs yet as adding new options to produce easier surroundings. typically every recreate of the automaton package is developed below a code name supported a course item. Past updates enclosed cake and doughnut. The most recent discharged versions of automaton are: **(Eclair)** that revamped the programme and introduced HTML5 and Exchange ActiveSync a pair of.5 support. that introduced speed enhancements with JIT improvement and therefore the Chrome V8 JavaScript engine, and supplementary Wi-Fi hotspot tethering and Adobe Flash support. **(Gingerbread)** that refined the programme, improved the soft keyboard and copy/paste options, and supplementary support for close to Field Communication. **(Honeycomb)** a tablet-oriented unleashes that supports larger screen devices and introduces several new programme options, and support multi core processors and hardware acceleration for graphics. The Honeycomb SDK has been discharged and therefore the initial device that includes this version, the Motorola Xoom pill, went on sale in Feb 2011. Google has chosen to withhold the event ASCII text file, that calls into question the "openness" of this automaton unleash.

Google claims this can be done to eliminate makers golf stroke a tablet-specific OS on phones, very like the previous time of year, wherever pill makers place a non-tablet optimized phone OS (Android a pair of.x) on their Tablets leading to unhealthy user experiences. **(Ice Cream),** a mix of cake and Honeycomb into a "cohesive whole. This version had new options supplementary to the Smartphone's like exposure enhancements, offline email looking out, biometric authentication unlock, network knowledge, and usage observation. SSL: The Secure Sockets Layer (SSL) and its successor, Trans-port Layer Security (TLS), area unit cryptanalytic protocols that were introduced to safeguard network communication from eavesdropping and change of state. to determine a secure affiliation, a consumer should firmly gain access to the general public key of the server. In most client/server setups, the server obtains associate degree X.509 certificate that contains the server's public key and is signed by a Certificate Authority (CA). once the consumer connects to the server, the certificate is transferred to the consumer. The consumer should then validate the certificate. However, validation checks aren't a central a part of the SSL and X.509 standards. Android Security: The open nature of automaton and its massive user base have created it a beautiful and profitable platform to attack. Common exploits and gear kits on the OS are often used across a good variety of devices, which means that attackers will perform exploits en bloc and re-use attack vectors. Google did take measures within the development of the Android kernel to create security measures in; the OS is sandboxed, preventing malicious processes from crossing between applications. While this plan to eliminate the idea of infection is admirable in some regards, it fails to deal with the problem of infection altogether. Android could be a victim of its own success, not simply within the means it's attracted malicious attention, however in its terribly nature. one among the explanations the OS has succeeded in gaining market share thus quickly is that it's open source; it's basically free for makers to implement. to boot this has light-emitting diode to substantial fragmentation of automaton versions between devices and implies that vendors are reluctant to roll-out updates, presumptively out of some concern relating to driving demand for future devices. Service: A Service is code that's durable and runs while not a UI. an honest example of this can be a media player enjoying songs from a play list. during a media player application, there would most likely be one or a lot of activities that permit the user to decide on songs and begin enjoying them. However, the music playback itself mustn't be handled by associate degree activity as a result of the user can expect the music to stay enjoying even once navigating to a brand new screen. during this case, the media player activity may begin a service victimisation Context. Start Service () to run within the background to stay the music going. The system can then keep the music playback service running till it's finished.

Features:

Storage: SQLite, a light-weight computer database, is employed for knowledge storage functions.

Connectivity: humanoid supports property technologies as well as GSM EDGE, IDEN, CDMA, EVDO, UMTS, Bluetooth, WI-Fi, LTE, NFC and Badger State easy lay.

Messaging: SMS and MMS square measure accessible styles of electronic communication, as well as rib text electronic communication and humanoid Cloud to Device electronic communication (C2DM) and currently increased version of C2DM, humanoid Google Cloud electronic communication (GCM) is additionally a region of humanoid Push electronic communication services:-

Multiple language support: humanoid supports multiple languages.

internet browser: the net browser accessible in humanoid is predicated on the ASCII text file internet Kit layout engine, not to mention Chrome's V8 JavaScript engine. The browser scores 100/100 on the Acid3 check on humanoid four.0.

Java: Java support whereas most humanoid applications square measure written in Java, there's no Java Virtual Machine within the platform and Java computer memory unit code isn't dead. Java categories square measure compiled into Dalvik executables and run on Dalvik, a specialised virtual machine designed specifically for humanoid and optimized for powered mobile devices with restricted memory and hardware. J2ME support may be provided via third party applications.

Multi-touch: humanoid has native support for multi-touch that was at first created accessible in handsets like the HTC Hero. The feature was originally disabled at the kernel level (possibly to avoid infringing Apple's patents on touch-screen technology at the time). Google has since discharged associate degree update for the Nexus One and therefore the Motorola Droid that permits multi-touch natively.

Bluetooth: Supports A2DP, AVRCP, causation files (OPP), accessing the phone book (PBAP), voice dialing and causation contacts between phones. Keyboard, mouse and joystick (HID) support is offered in humanoid three.1+, and in earlier versions through manufacturer customizations and third-party applications.

Tethering: Humanoid supports tethering, that permits a phone to be used as wireless/wired Wi-Fi hotspot. Before humanoid a pair of.2 this was supported by third-party applications or manufacturer customizations.

Screen capture: humanoid supports capturing a screenshot by pressing the ability and volume-down buttons at constant time. before humanoid four.0, the sole ways of capturing a screenshot were through manufacturer and third-party customizations or otherwise by employing a computer affiliation (DDMS developer's tool). These various ways square measure still accessible with the newest humanoid.

3. DESIGN OF HUMANOID SOFTWARE

Android software could be a stack of computer code parts. Main parts of humanoid software design or computer code Stack square measure Linux kernel, native libraries, humanoid Runtime, Application Framework and Applications.

3.1. Linux Kernel

Linux Kernel (Linux a pair of.6) is at rock bottom layer of the computer code stack. Whole humanoid software {is built is created is constructed} on this layer with some changes made by the Google [5]. Like main software it provides the subsequent functionalities: method management, Memory Management, device management (ex. camera, keypad, show etc). humanoid software interacts with the hardware of the device with this layer [6]. This layer additionally contains several necessary hardware device drivers. Linux kernel is additionally chargeable for managing computer memory, networking, drivers, and power management [7].

3.2. Native Libraries Layer

Native Libraries Layer On the highest of the Linux Kernel layer is Android's native libraries. This layer permits the device to handle differing types of knowledge. Knowledge is restricted to hardware. of these libraries square measure written in c or c++ language. These libraries square measure known as through java interface. Some necessary native libraries are: Surface Manager: it's wont to manage show of device. Surface Manager used for composing windows on the screen. SQLite: SQLite is that the information utilized in humanoid for knowledge storage. It is computer database and accessible to any or all applications. Web Kit: it's the browser engine wont to show hypertext mark-up language content. Media framework: Media framework provides playbacks and recording of varied audio, video and movie formats.(as an example MP3, AAC, AMR, JPG, MPEG4, H.264, and PNG).

3.3. Humanoid Runtime

Humanoid Runtime consists of Dalvik Virtual machine and Core Java libraries. it's placed on constant level because the library layer [5]. Dalvik Virtual Machine could be a variety of Java Virtual Machine used for running applications on humanoid device. The Dalvik VM permits each humanoid application to run in its own method, with its own instance of the Dalvik virtual machine. The Dalvik VM permits multiple instance of Virtual machine to be created at the same time providing security, isolation, memory management and threading support [8]. not like Java VM that is process-based, Dalvik Virtual Machine is register-base. Dalvik Virtual Machine run dex files that square measure created from .class file by dx tool. dx tool is enclosed in humanoid SDK. DVM is optimized for low process power and low memory environments. DVM is developed by Dan Bornstein from Google [9]. 2.4 Application Framework the appliance Framework layer provides several higher-level services or major genus Apes to applications within the type of Java categories. Application developers square measure allowed to form use of those services in their applications [6]. These square measure the blocks with that developer's applications directly act. Necessary blocks of Application framework are: Activity Manager: It manages the life cycle of applications. Content Providers: it's wont to manage the information sharing between applications, manages a way to access knowledge from different applications. telecom Manager: it manages all voice decision connected functionalities. Location Manager: it's used for Location management, mistreatment GPS or cell tower. Resource Manager: Manage the assorted sorts of resources utilized in Application [8]. 2.5 Application Layer The Applications Layer is that the prime layer within the humanoid design. Some applications come back pre-installed with each device, such as: SMS shopper app, Dialer application programme and phone manager. A developer will write his own application and might replace it with the prevailing application [8].

4. COMPLETELY DIFFERENT SAFETY FEATURES OF HUMANOID OS

Android software ought to make sure the security of users, user's knowledge, applications, the device, and therefore the network. to attain the safety of those parts humanoid provides these key safety features [10]:

1) Security at the software level through the Linux kernel.

2) Application sandbox for all applications

3) Secure inter-process communication.

4) Application sign language.

5) Application-defined and user-granted permissions.

Linux Kernel humanoid software is predicated on Linux kernel. as a result of its open supply nature it's researched, attacked and glued by several analysis developers. therefore Linux has become stable and secure kernel. Linux kernel provides humanoid with many key safety features including:

a) A user-based permissions model within the Linux classification system every file and directories have 3 user based mostly permissions. Owner group different user owner the Owner permissions apply solely the owner of the file or directory. Cluster - The cluster permissions apply solely to the cluster that has been appointed to the file or directory. Different users - {the different the opposite} Users permissions apply to any or all other users on the system. Every file or directory has 3 basic permission types: scan - The scan permission means that user's ability to scan the contents of the file. write permissions mean's user's ability to write down or edit a file or directory. Execute the execute permission means that user's ability to execute a file or read the contents of a directory [11]. This permission model ensures that correct security is maintained whereas accessing humanoid files.

b) Method isolation: The humanoid software assigns a novel user ID (UID) to every humanoid application and runs it as a separate method.

c) Protrusive mechanism for secure IPC.

d) The power to get rid of redundant and insecure components of the kernel [10].

4.1. The appliance Sandbox

A sandbox could be a security mechanism for separating running programs and limiting the resources of the device to application. it's usually wont to execute untested code or programs from entrusted users and entrusted websites. By mistreatment sandboxing technique restricted access to device's resources is given. thus security of the system is increased . Sandboxing technology is often wont to check unproven programs which can contain a deadly disease or different malware code, while not permitting the computer code or code to hurt the host device. With the assistance of sandbox entrusted program access solely those resources of the device that permission is granted. Permission is denied if it tries to access different resources of the device [12].

4.2. Secure inter-process communication

Some of the applications still use ancient Linux techniques like network sockets, classification system and shared files for inter-process communication. However humanoid software additionally provides new mechanism for IPC like Binder, Services, Intents and Content Providers. of these mechanism permits developers to verify the identity of application and additionally wont to set the safety policies [13].

4.3. Application sign language

In order to put in and run applications on humanoid OS they have to be digitally signed. With this mechanism humanoid OS distinguishing the author of associate degree application. This feature additionally won't to establishing trust relationship between applications. If associate degree application isn't any signed properly then it cannot be put in on the mortal additionally. Some commonplace tools like Key tool and Jar signer square measure wont to generate keys and sign application .apk files [15]. 3.5 Application-defined associate degree user-granted permissions square measure an humanoid security mechanism to permit or limit application access. By default, humanoid applications haven't any permissions granted, creating them safe by not permitting them to realize access to protected genus Apes [14]. a number of the protected genus Apes include: Camera functions, Location knowledge (GPS) ,Bluetooth functions, telecom functions, SMS/MMS functions and Network or knowledge connections. These resources square measure accessed solely through the software [10].

5. CONCLUSION

From higher than discussion it's clear that humanoid software follows a spread of security mechanism. Once developer can install associate degree application a brand new user profile thereupon application is formed. Every application can run with its own instance of Dalvik VM. Therefore applications cannot access every other's knowledge. If applications need to access shared knowledge or resources then they need permissions. All humanoid applications square measure signed therefore users understand that the appliance is authentic. The sign language mechanism permits developer to manage that applications will grant access to different application on the system.

REFERENCES

[1] http://www2.dcsec.unihannover.de/fies/android/p50-fahl.pdf

[2] http://digitalforensicssolutions.com/papers/android-memory-analysis-DI.pdf

[3] http://www.uandistar.org/2011/06/paper presentation-on-android.html

[4] http://www.studymode.com/essays/Android Research-Paper-1068648.html

[5] http://www.4shared.com/office/0RX_5-iE/file.html

[6] http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/WIKIPEDI/W110410O.pdf

[7] http://students.mint.ua.edu/~pmkilgo/etc/android-os.pdf

[8] http://www.acumin.co.uk/download_files/WhitePaper/android_white_paper_2.pdf

[9] http://ptcoresec.eu/2013/05/02/part-1-getting -to-know-android/

[10] http://source.android.com/devices/tech/security/

[11] http://www.linux.com/learn/tutorials/30952 understanding-linux-file-permissions

[12] http://en.wikipedia.org/wiki/Sandbox_(computer_security)

[13] http://developer.android.com/training/articles/security-tips.html

[14] http://www.ibm.com/developerworks/library/x-androidsecurity

[15] http://developer.android.com/tools/publishing/app-signing.html

BIOGRAPHIES

Nilesh N. Chawande
M.E - (Computer science and Engineering)
Amravati, Maharashtra, India.
NileshChawande@gmail.com