International Journal of Innovative Technology & Adaptive Management (IJITAM)
ISSN: 2347-3622, Volume-1, Issue-8, May, 2014

An Overview on Android Security Threats And Protection

Rashmi Kumari, B-Tech 3rd year, Computer Science and Engineering, reshu.rashmikumari@gmail.com, C.V.Raman College of Engineering, Bhubaneshwar, Odisha.

Akash Kumar Athghara, B-Tech 3rd year, Electronics and Telecommunication Engineering,akash.athghara@gmail.com,Veer Surendra Sai University of Technology,Burla,Odisha.

Amit Kumar Gupta, B-Tech 3rd year, Computer Science and Engineering,amy.amitgupta@gmail.com,C.V.Raman College of Engineering,Bhubaneshwar,Odisha.

Dipankar Pramanik, Assistant professor, CSE, AIET, prdipu@gmail.com,Bhubaneswar, ORRISA.

Abstract.

The number of Androidbasedsmartphones is growing rapidly. They areincreasinglyused for securitycriticalprivate and business applications, such as onlinebanking or to access corporate networks. This makes them a very valuable targetfor an adversary. Up to date, significant or largescaleattacks have failed,but attacks are becoming more sophisticated and successful. Thus, security is ofparamount importance for both private and corporate users. In this paper, wediscuss Android security and presentour extensible automated exploit execution framework. First, we provide a summaryof the Android platform, current attack techniques, and publicly knownexploits. We discuss how malware can propagate to Android smartphonestoday and in the future, and which possible threats arise and security measures to cope with all the threats.

Keywords: Android security, smartphone security, malware.

1. Introduction

Android [1] [2] is a modern mobile platform that is designed to be trulyopen source. Android applications can use advanced level ofhardware and software, as well as local and server data, exposedthrough the platform to bring innovation and value to consumers.

Android platform must have security mechanism to ensuresecurity of user data, information, application and network.

Open source platform needs strong and rigorous securityarchitecture to provide security. Android is designed with multilayeredsecurity that provides flexibleness needed for an openplatform, whereas providing protection for all users of theplatform designed to a software stack, android includes anoperating system, middleware and core application as acomplete. Android powers hundreds of millions of mobiledevices 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 simply work on versatile security

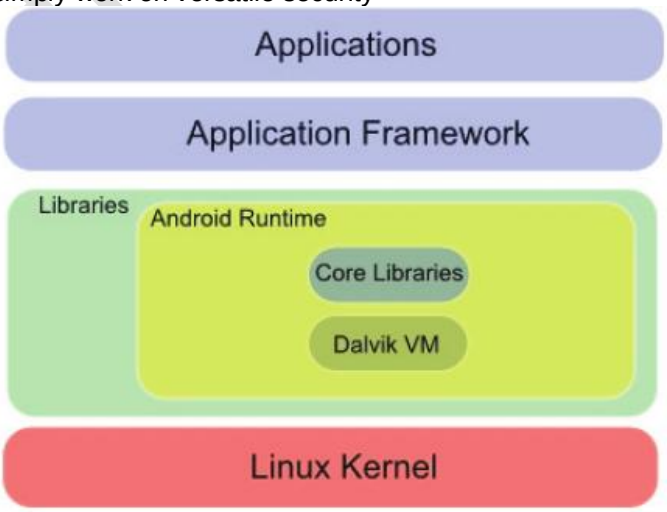

Figure 1. Android Architecture

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 thoseapplications.

2. Android and Android Security

Mobile operating systems preinstalled bon all currently sold smartphones need to meet different criteria than desktop and server operating systems, both in functionality and security [3]. Mobile platforms often contain strongly interconnected, small and lesswell controlled applications from various single developers, whereas desktop and

server platforms obtain largely independent software from trusted sources. Also, users typically have full access to administrative functions on non mobile platforms. Mobile platforms, however, restrict administrative control through users. As a consequence, different approaches are deployed by the Android platform to maintain security.

2.1. Android Platform

The Android operating system is illustrated in Figure 2. Apps for Android are developed in Java and executed in a virtual machine, called Dalvik VM. They are supported by the application framework, which provides frequently used

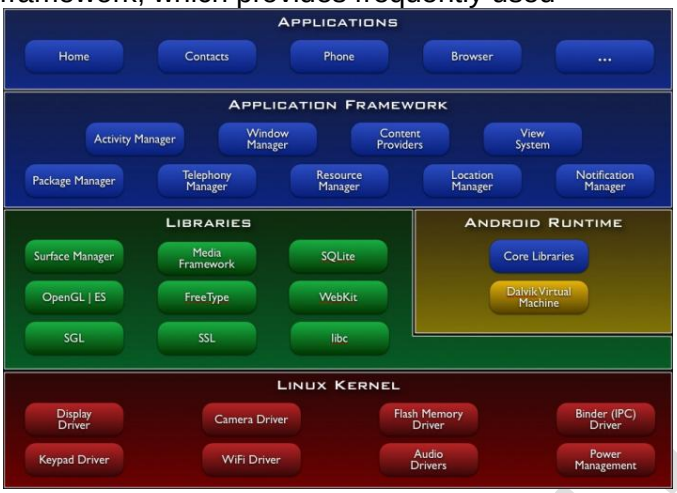

Figure 2: Android System architechture

functionality through a unified interface. Various libraries enable apps to implement graphics, encrypted communication or databases easily. The Standard Library (“bionic”) is a BSDderivedlibc for embedded devices. The respective Android releases’ kernels are stripped down from Linux 2.6 versions. Basic services such as memory, process and user management are all provided by the Linux kernel in a mostly unmodified form.

2.2 File System and User/Group Permissions

As in any Unix/Linuxlike operating system, basic access control is implemented through a threeclass permission model. It distinguishes between the *owner* of a file system resource, the owner’s *group* and *others*. For each of these three entities, distinct permissions can be set to read, write or execute. This system provides a means of controlling access to files and resources. For example, only a file’s owner may write (alter) a document, while members of the owner’s group may read it and others may not even view it at all.

2.3 Android API Permission Model and Manifest File

On installation, the user is presented with a dialog listing all permissions requestedby the app to be installed. These permission requests are defined inthe Manifest File AndroidManifest.xml, which is obligatory for shippingwith every Android app.
However, this system has a few flaws:
• *All or nonepolicy*
• Often, users cannot judge the appropriateness of permissions for the appin question
• Circumvention: Functionality, which is supposed to be executable onlygiven the appropriate permissions, can still be accessed with fewer permissionsor even with none at all.

2.4 Android Market (”Google Play”)

As anyone can publish an app after registration as a developer for USD25 anddue to its availability to all Android users, the Android Market, recently renamed”Google Play”, was and is the main channel of malware distribution.
The majority of all infections is conducted through free illegitimate copies of paidcontent. Users unwilling to pay for such content turn to pirated copies, which areoften altered to deliver malicious code. This process is known as “repackaging”.

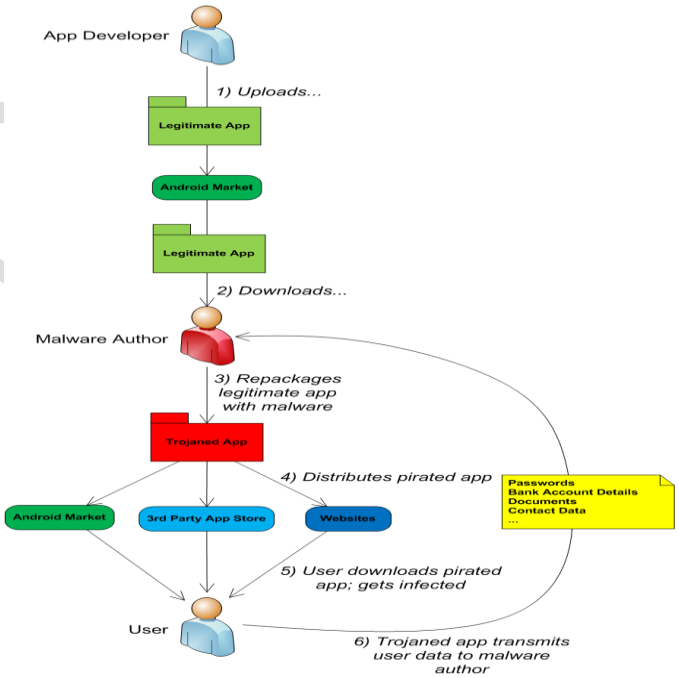

Figure 3:Repackaging Process

2.5 Remote Installation and Uninstallation

Google possesses the ability to remotely remove or install any app from/to allAndroid devices. It is used for the removal of malicious applications and theaddition of new services to phones which may require installation of new apps.

2.6 Patch Process

The Android OS patch process is a complex and timeconsumingcombined effortwith many parties involved. After vulnerability has been discovered in an AndroidOS component, Google has to develop and push out a patch. This patchis merged with the Android OS source code, which is then provided to

devicemanufacturers. They adjust the new code base to their devices, affecting notonly original equipment, but also handsets specifically produced for carriers orgeographical regions.

2.7 SEAndroid

The United States' National Security Agency has recently announced the commencementof the SEAndroid (Security Enhanced Android) project as an additionto the Android kernel [10].

2.8 Apps and Native Code

Android apps are developed in Java and executed inside the Dalvik Virtual Machine,a registerbasedJava Virtual Machine. Each app receives its own DalvikVM instance and user ID, hence separating process memory and files throughmeans of the Linux kernel

3. Android Platform Security Architecture

Android seeks to be the most secure and usable operating systemfor mobiles by re-purposing classical operating system securitycontrols to protect user data, system resources and provideapplication isolation.
Android provides following security features to achieve theseobjectives are first robust security at the operating system levelthrough the Linux kernel, second compulsory application sandboxfor all applications, third secure interposes communication, fourthapplication signing, and sixth application defined permission [3] [4] anduser have to grant permissions. Android operating system code running as root, all process runabove the Linux Kernel is restricted by the ApplicationSandbox.

3.1. Security in Android:

i. Android is open source platform,developers will work along to enhance it.
ii. Android platformis multitasking software; therefore no application will gaincritical access to the components of OS.
iii. Android platform isUNIX based operating system that is the most secure operatingsystem.
iv. The developers need a unique signature to publishtheir application on market[4].
v. Users will report a possiblesecurity flaw through their Google account.
vi. All applicationson android need permission from the user at the time ofinstallation.

4. Vulnerabilities

Taking specific malware out of the equation, what are some of the threats/vulnerabilities on Android devices thatmight be cause for concern? These certainly are not comprehensive, but do cover a significant range of the vulnerabilitiesand risks that may be exploited on the Android OS:

4.1. User as admin

Install apps, grant app permissions, download data, and access unprotected networks - The user can reign free overtheir Android domain without restriction.

4.2. The Android Market

Google's verification processes for applications entering their market have been shown to be woefully lacking overthe last year or two, leading to a number of malware-infected apps and games being made legitimately available tousers.

4.3. Gateway to PC

HTC devices have long been able to utilise a VPN, but increasingly other applications are becoming available forremote access – Go to Meeting, TeamViewer, Remote Rackspace. Although secured, these third party services stillprovide a line in to the corporate network and may be implemented fairly easily on to an endpoint.
Any Android device can be connected to a PC via a USB cable, laying out the contents of its SD card for read/write/delete. The SD card itself as removable storage can be easily accessed directly as well. Indeed these methods couldbe utilised themselves for bringing malware in to a corporate network, for downloading malicious content on to aPC or sucking up data as soon as it is connected.

4.4. Application permissions

In the form of a pop up, the user may see these notifications as a nuisance, a delay in accessing the newlydownloaded Angry Birds levels. Or they may simply not understand the nature of the requests. Commonpermissions that may (read: should!) raise an eyebrow would include 'Read/Send SMS', 'Access Fine Location', 'AccessIMEI, phone identity', 'Brick' (required to disable the device in trace and wipe apps), 'Access camera',and so on. Such requests may be integral to functionality, but could equally be recording calls and transmittingsign-in credentials.

4.5. Malicious application injections

Data/process transfers between virtualised application environments are handled by a protocol of implicit and explicitintents. Transmission or interception of an intentby a malicious application can result in data being compromisedas the target app will respond to the string, potentially resulting in data loss.

4.6. Third party applications

One of the great things about Android is choice in terms of standard functionality, such as address books,messaging, keyboards, etc. In a rapidly growing
OS environment it can be difficult to identify reputable vendors. Both Facebook[8] and Twitter[9] transmit mobile app datain the clear, i.e. without encryption, on nearly all devices. This happens

despite the development of such securitymeasures for web app versions.

4.7. Rooting

Rooting an Android device is akin to jail-breaking aniPhone, it opens out additional functionality and servicesto users. The process of gaining root access, requires the device to be switched from S-On to S-Off (where S =security). Additionally, root is a common exploit used by malicious applications to gain system-level access to yourAndroid.
DroidKungFu is one such threat that can root a system and install applications at that level, it escapesdetection by utilising encryption and decryption to deliver a payload.

4.8. Wi-Fi

The vulnerability [5] of Android devices running 2.3.3 to compromise on unprotected Wi-Fi networks apparently cameas a surprise to many, it shouldn't have, when is this practice ever safe?! Beyond highlighting the need for betterconsumer security awareness, it leads to some other considerations around secure Wi-Fi access. Ideally sign incredentials should always be completed over a secured network, but sometimes this isn't enough. FaceNiff is aneasily downloadable application that allows the user to intercept the social networking logins of any Android ontheir network. The only way this exploit won't work is if the user is utilising SSL. Furthermore, devices running 2.3(or rooted older devices) can act as a Wi-Fi hotspot – as an Information Security Manager, how happy would you beabout unverified users and devices connecting to a smartphone with a corporate footprint?

4.9. Remote installation

All versions of Android [6] were at one point vulnerable to the remote writing of malicious JavaScript to the SD cardthrough accessing an infected web page– an html download does not prompt for user confirmation on the OS.

4.10. Privacy

This is the primary issue with Android as a consumer device –functionality over security. Other applications claiming localised services could utilise GPS permissions for locationtracking.

4.11. Manufacturer trust

Whatever their intentions, manufacturers play a significant role in user privacy. Uncovered recently is anapplication that sits at root level on new HTC devices (Evo, Evo 3D, Thunderbolt, Sensation) which collectsand transmits a range of information on users including accounts, phone numbers, SMS, system logs, GPSlocations, IP addresses, and installed apps[14]. It is bad enough that HTC feel it is appropriate to collect and usethis data without notifying the user, it is even worse that it failed to secure it! Consequently any app withthe 'Access internet' permission can access this data.

5. Security Measures

Some of thecontrols and mitigation processes we can take to protect the device and the information it may access. Some of thisadvice could be implemented by best practice, policy, or a user toolkit within the enterprise, but roll out to individualuser devices should also be considered.

5.1. Root and S-off

Rooting a device allows the user to exercise control over it. Upon completion of theroot process, the flashed Clockwork Recovery Mod will install the SuperUser application. SuperUser notifies on allrequests for root access, asking for authentication of the process. Immediately threats such as DroidKungFu becomeless intimidating, it can do very little at a root level if that root is controlled and monitored.

5.2. Permissions management

Many of the attack vectors [7] and vulnerabilities which might be considered feasible on an Android device, rely onbeing able to overcome the OS sandboxing.
LBE Privacy Guard acts as somewhat of an application firewall and can solve this problem by granting the user theability to block an application's individual permissions. I might be happy for a game to have internet access, but Ican then block other permissions I feel unnecessary such as sending SMS or reading my phone number. The applicationwill only work on rooted devices, but is one of the most effective security measures you can take on Android.
You can set permissions, and will be notified of any requests that you haven't already specified a preference for.

5.3. Trace and wipe

If your Android device is lost or stolen, you can usethese applications to remotely ping the device for its location and/or instruct it to delete specific content. Taking out some of these 'traceand wipe' apps really is as easy as uninstalling them from the Application Management menu. Any solutions onthat basis should be manufacturer implemented or be at root level, e.g. TheftAware, to prevent them from beingbypassed.
TheftAware also does the obvious and conceals its identity with an innocuous name designated by the user; thehidden application remains visible but concealed until it is activated, at which point it disappears from the front endaltogether.

5.4. Device locking

The reason for implementing some level of screen lock security on a smart phone should be pretty obvious to eventhe most ley user. Such a mass of personal data and material simply cannot be left on a table, available to all whoare willing to swipe to unlock.

5.5. Securing data

Firstly all the usual security rules need to be exercised, e.g. only connecting to trusted, secure networks. Beyond thisthough, the user needs to consider what the appropriate security measure is for whatever they're doing –encryption for data transmission, VPN for secure connection.
WhisperCore provides full desk encryption and also retains an encrypted backup of all data on the device.

5.6. Installing trusted packages

Given the ability to install non-Market applications on to a Google device off the bat (unlike iPhone wherejailbreaking is required) [2][4] , and the recent spate of malicious apps on the Android Market, it might be time to considerverifying contents of APK files. For the uninitiated, APK files are the standard Android install file format and are a variantof JAR. A program called APK Inspector has recently been released that will scan the assets, resources, and certificatescontained within the APK to ensure it is secure. This is available via http://code.google.com/p/apkinspector and could beutilised as part of the application install process to ensure all packages are verified before being flashed to the phone.

5.7. Anti-virus

Anti-virus must as part of a comprehensive security solution.The lack of malware signatures, hardware limitations, and an uncertain market potential in the early days ofAndroid meant that the big players were initially reluctant to enter in to the space with bespoke solutions.

5.8. Authentication

Much of the mention of authentication around Android comes in the form of using the devices as a means ofauthentication, like an OTP or token. What is often overlooked is authentication on the device itself.

5.9. Link security

Users love to click links. There are a number of vendors that have created link security applications for PC, and there areoptions available on Android too, albeit in fewer numbers. AVG with its Mobilation app and Lookout MobileSecurity seem to be leading the way here, providing a solution for secure browsing.

5.10. Email security and mobile device management

Secure Active Exchange sync on the default mobile app, or encourage the use of a web service such as Mimecastor OWA. Automatically use an encryptedSSL connection for any transfer of email data. This is the only way to comply to the security protocols onExchange, without sending encrypted messages through a VPN using another application. Mobile Device Management (MDM) and Mobile Email Gateway (MEG) products from the likes of GoodTechnology, AirWatch, and Echoworx allow enterprises to roll out email security solutions that are compliant toindustry standards such as SOX and HIPAA.

5.11. Firewalls

Really LBE PrivacyGuard should be helping in scratching that firewall itch on Android, but the data that istransmitted from your smart phone should be monitored as well. This may seem superfluous but given that iPhoneusers' location data is tracked and sent unencrypted via iTunes, this might not seem like such an unnecessarymeasure. Whisper Core Systems, which brought FDE to Android, also offers Whisper Monitor.

5.12. Secure storage and NFC payments

There are numerous applications available via the Market offering secure wallets/folders and passwordmanagement. This concept offers a theoretically secure and password-protected environment for sensitiveassets. However there is always the issue of vendor trust when it comes to utilising such applications to protectinformation. Of course when Google Wallet rolls out properly to more than one Android device (i.e. Nexus S)
it should theoretically prove a logical solution to secure storage, however it will also open up the potentialimplications of a security breach or lost device significantly. Losing a smartphone is already cause enough forconcern, but going forwards there will be more significant financial implications, as the emergence of NFC-enableddevices will mean that this is now akin to losing a debit card or wallet.

6. Conclusion

Aspointed out by recent research and publications, however, They are now capable ofspreading mechanisms which do not require explicit user confirmation. Malwaremay be delivered unnoticed through desktop computers, other Android devicesor trojanized apps.
Due to the open usage model of the Android market, malicious apps cannot beavoided

completely. Especially pirated apps or multimedia content in popular demandtargeting user groups with typically low awareness levels are predestined tospread to many devices before being identified by Google as malware.

In this paper, we can conclude that as Android is an open source platform, attacks on Androidpowered devices are becoming more sophisticated. So, anyone should take the above mention security measure to prevent those attacks.

7. Future Work

In future, we will study the upcoming android treats and their characteristics, their functionality over Android OS. Basis on this, we will also try to provide the security measures and protection to the Smartphones.

References

1. Android Open Source Project. Android Security Overview.http://source.android.com/devices/tech/security/index.html.(2013)
2. Tiwari Mohini, Srivastava Ashish Kumar and Gupta Nitesh ,“Review on Android and Smartphone Security”,isca, Vol. 1(6), 12-19, November (2013).
3. Android Security Team, “Android Security Overview,” December 2011.
4. H. Lockheimer, “Android and Security,” February 2012.
5. J. Oberheide, “remote kill and install on google android,” June 2010.
6. T. Vidas, D. Votipka and N. Christin, “All your droid are belong to us: A survey of current android attacks,” in *5th USENIX Workshop on Offensive Technologies*, Carnegie Mellon University, August 2011.
7. C. Mullaney, “Android.Bmaster: A MillionDollar Mobile Botnet,” February2012.

www.ijitam.org