Data security on Android OS Against Malware

Ms. Kanchan Sanjay Patil
ASM INSTITUTE OF MANAGEMENT & COMPUTER STUDIES (IMCOST), THANE, MUMBAI.
University Of Mumbai

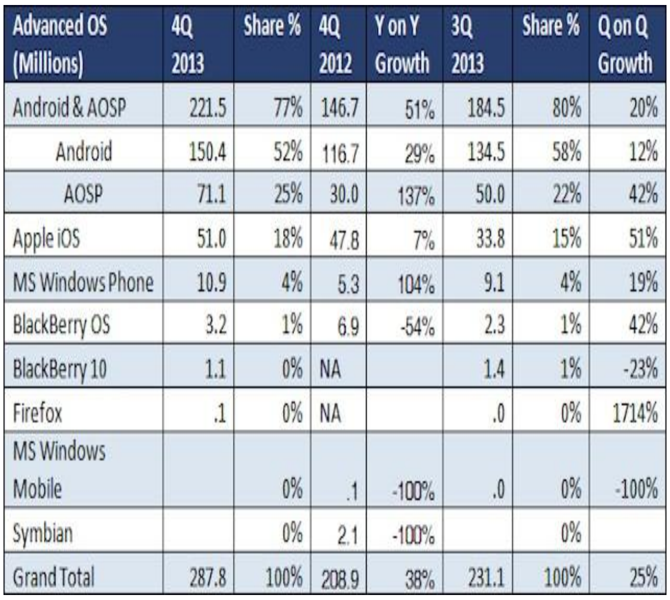
Mr.Vinod Shivdas Panchal
ASM INSTITUTE OF MANAGEMENT & COMPUTER STUDIES (IMCOST), THANE, MUMBAI.
University Of Mumbai

Abstract - **Worldwide number of Android users are growing rapidly. Android OS used in smartphone itself but now it comes in PC, Tablets, TVs. This various ways makes them free to access internet by different conditional applications. Which increases security threats in private and business applications, such as online banking or to access corporate networks. Now a days intruders becomes attackers and attacks are becoming more sophisticated and successful. Thus, security is of pre-eminent importance for both private and corporate users. In this paper, we are going to introduce status of today's Data security on Android OS and introduce our framework, which is extensible automated exploit execution. We discuss about malware which can propagate today and also possible threats arise in the future. For example, device to device infections are possible if physical access is given. In this research, we first introduce summary of Then, we introduce the Android platform, current attack techniques and publicly known exploits our extensible exploit execution framework which is capable of performing automated vulnerability tests of Android smartphones. It incorporates currently known exploits, but can be easily extended to integrate future exploits. In this paper, discuss the Data security on Android OS against Malware**

Keywords* – *SEAndroid project, Android , Android OS, Android devices, Data security, Mobile TAN, malicious applications, Dalvik VM in security, AppArmor

I. INTRODUCTION

Recent years, the sales and use of smartphone in handheld device rapidly increases. Android operating system among all OS led by the Open Handset Alliance, likely led by Google Inc., which is market dominating. In Q4 2013, 77% of all devices sold were Android devices, followed by Apple's iOS (18.0%), and MS Windows Phone (4%) according to Gartner analysis[1].



Advanced OS (Millions)	4Q 2013	Share %	4Q 2012	Y on Y Growth	3Q 2013	Share %	Q on Q Growth
Android & AOSP	221.5	77%	146.7	51%	184.5	80%	20%
Android	150.4	52%	116.7	29%	134.5	58%	12%
AOSP	71.1	25%	30.0	137%	50.0	22%	42%
Apple iOS	51.0	18%	47.8	7%	33.8	15%	51%
MS Windows Phone	10.9	4%	5.3	104%	9.1	4%	19%
BlackBerry OS	3.2	1%	6.9	-54%	2.3	1%	42%
BlackBerry 10	1.1	0%	NA		1.4	1%	-23%
Firefox	.1	0%	NA		.0	0%	1714%
MS Windows Mobile		0%	.1	-100%	.0	0%	-100%
Symbian		0%	2.1	-100%		0%	
Grand Total	287.8	100%	208.9	38%	231.1	100%	25%

Figure 1.1 Mobile sale 4th quarter [2]

Most of areas are covered by Android smartphones, usually both in private and public related work areas. So securing These devices are at high approach by user. Users use their smartphones to perform daily tasks ranging from everyday communication with friends and family to the management of banking accounts and accessing sensitive work related data. So all these factors, combined with limitations in Main access of device control through owners and security applications like the Mobile TAN for Money transactions. This is a main reason to make Android smartphones a very attractive target for attackers and malware authors.

II. ANDROID AND ANDROID SECURITY

All smartphones and apk desktops needs to meet different criteria than desktop and server operating systems both in functionality and security. Sometimes Mobile platforms contain strongly interconnected, small and less-well controlled applications from various single developers, whereas desktop and server platforms obtain largely independent software from trusted sources. Typically user can have privileges to access any function on non-mobile platforms also he can restrict on privileges. Different steps has been developed by android developers to maintain security.

a. Apk Platform

The Android operating system is illustrated in Figure 2.1. Dalvik VM is virtual machine where apps developed in java have been executed. This framework which support application and provides unified interface to provide frequently used functionality. Some libraries enables encrypted communications database or implemented graphics. Embedded devices have some Standard libraries known as Bioninc / BSD-derived libc. The respective Android releases' kernels are stripped down from Linux 2.6 versions. Basic Services like memory process and user management in unmodified form are provided by Linux kernel.

However, for several Android versions, the deployed kernel's version was already out of date at the time of release. This has lead to a strong increase in vulnerability, as exploits were long publicly available before the respective Android version's release.

Detailed information on all security features can be retrieved from the Android Security Overview [2] on the official Android Open Source Project website.

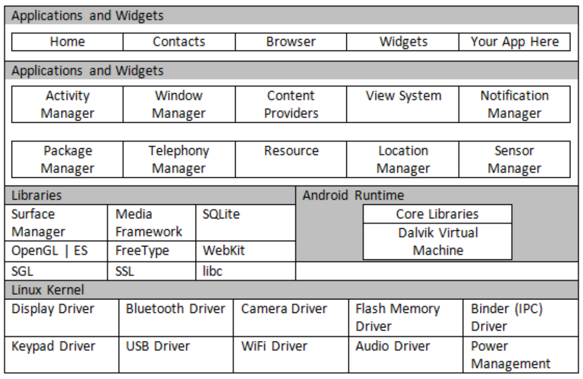


Figure 2.1: Android System Architecture

b. File System and User/Group Permissions

Three class permission model helps to implements basic access control in any Unix/Linux like OS and they are as It distinguishes between the owner of a file system resource, the owner's group and others. There are 3 distinct permissions read, write and execute can set to those three entities, which provides access on files and resources.

In traditional desktop and server environments, many processes often share the same group or even user ID (namely the user ID of the user who started a pro-gram). As a result, they are granted access to all files belonging to the other pro-grams started by that same ID. In traditional environments with largely trusted software sources this may suffice, though Mandatory Access Control approaches trying to establish a more fine-grained model do exist, such as AppArmor and SELinux [2].

However, for mobile operating systems this is not sufficient. Finer permission distinction is needed, as an open app market is not a strongly monitored and trustworthy software source. With the traditional approach, any app executed under the device owner's user ID would be able to access any other app's data. Hence, the Android kernel assigns each app its own user ID on installation. This ensures that an app can only access its own files, the temporary directory and nothing else – system resources are available through API calls. This establishes a permission-based file system sandbox.

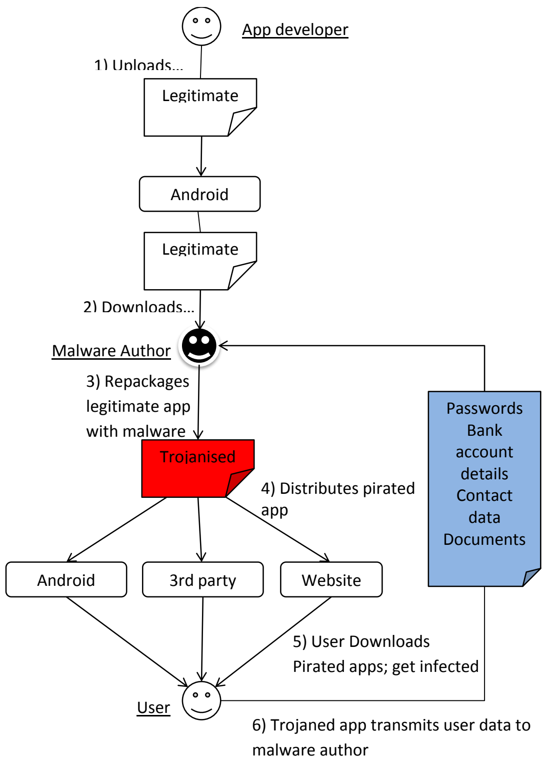

App developer
1) Uploads...
Legitimate
Android
Legitimate
2) Downloads...
Malware Author
3) Repackages legitimate app with malware
Trojanised
4) Distributes pirated app
Passwords Bank account details Contact data Documents
Android
3rd party
Website
5) User Downloads Pirated apps; get infected
User
6) Trojaned app transmits user data to malware author

Figure 2.2: Repackaging Process

c. ***Android Market ("Google play")***

Google Play itself is the main channel of malware distribution. As anyone can first registered as developer for USD25 and the publish an app.

free illegitimate copies of paid content are major reason of all infections. Human tendency to take free content instead of original paid content leads to take pirated applications which are often altered to deliver malicious code. This process, known as "repackaging", is illustrated in Figure 2.2.

An Android botnet uncovered latest employed this technique to infect several ten thousand devices, generating a profits of multiple million dollars per year through premium communication services .

Google recently gives service named "Bouncer" in Google test apps possibly malicious behaviour, Bouncer examines apps which is automatically submitted to the Android Market and by execution inside a virtual Android environment in Google's cloud infrastructure. If malicious code detect, manual analysis is performed to prevent false alarms.

Bouncer decreases numbers of malware apps on google play but this system does not provide security against modern attack approaches. It is still suspicious that 'is bouncer able to identify unknown or altered exploits ?'.

In Conclusion, Bouncer can prevent from malicious app in Google play. This includes apps which do not rely on dynamic and platform-specific privilege escalation exploits or implement all spying or remote control functionality in Java. Bouncer does not prove effective against more sophisticated attacks.

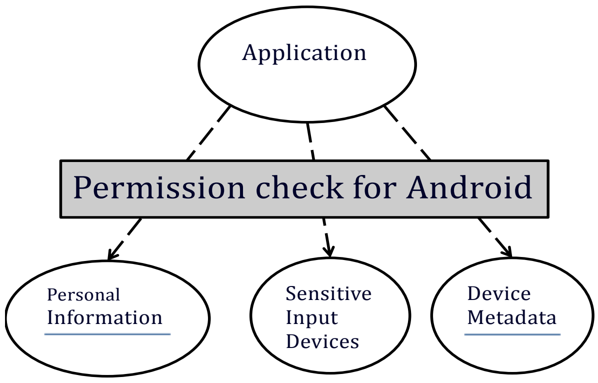


Figure 2.3

III. PROPOGATION SCENARIOS

For mobile malware scenarios significantly differ from those of desktop malware. Direct self-spreading mechanisms over primary communication networks known from desktop environments are very unlikely different approaches exist, which utilize existing infrastructure such as the Android Market and websites. Also, novel approaches such as PC-to-device and device-to-device infections will be discussed.

a . Malicious Applications

Trojanised programs on desktop and malicious apps on devices are similar scenario and when we compare them we found they provide high convenience for malware developers, as the Android Market and 3rd party app markets potentially give access to all Android users.

App Markets

Some paid contents on the Android market are very popular in users. Malware authors target those contents and make them channel for infection.

Famous application with paid versions theft by malware author. They repackage them with malicious contents and make them free of cost on various place like App Markets.

3rd-party app markets installed on modified operating system distributions ("custom ROMs") are particularly put at risk of containing malicious apps. Pirated versions of paid content gets many hits, likes and Downloads. So this version purposely get place in market by market operator. Also, 3rd-party app markets, if operated by a non-professionals, whom do not have the person to closely monitor app uploads. Malware recognition techniques such as Google's Bouncer are not deployed. These community run app markets do not require a registration fee as opposed to USD25 of the official market. malware author would be charged every time in the official market, so their one of accounts is removed for spreading malware.

Websites

Many users allow websites to install their apps on device. There are no eye watch on this case , which increasing the risk of installing trojanized apps. Also, users can be redirected to fake websites, when this option is activated, supposedly supplying a "critical update". Some smartphones are vulnerable to some attacks but Based on the user agent identification string of a device's browser, targeted attacks can be conducted. Rogue networks or attackers may even be able to rewrite web traffic to replace legitimate apps with malicious ones.

b. Infection via Personal Computers

Android operating system cannot be remotely exploits because its security model, which successfully prevents vulnerable, compromised apps from modifying any operating system components, device-to-device infections which is virtually impossible. This applies for all Android versions prior to Version 3.1, which features USB host mode. USB host mode can be used to infect other Android-based smartphones with USB debugging enabled. Versions prior to Version 3.1 account for around 90% of all Android devices as of May 2012.

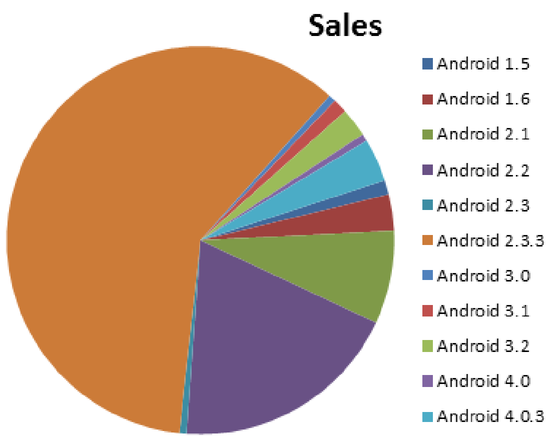


Figure 4.1: Android Version Distribution as of May 1, 2012 [3]

Technically, desktop computer malware have to implement the Android Debug Bridge's protocol to install arbitrary software on any device with USB debugging activated.

IV. CONCLUSION

Negligently behave over android usage by all Android users gives attackers more sophisticated ways to attack. This attackers do not required any explicit user confirmation as they developed that kind of mechanism says by recent research and publications. Malware can be infect other Android devices or trojanized apps through desktop computers.

Rookie users in Android is mostly targeted by malicious apps rather than Pirated apps market or multimedia. This apps infect them before Google identify them as malware.

Android get easily attacked by malicious apps or say trojanized apps. If any device stays enable in USB-debugging mode than attacker gets the privilege to attack the target easily that can be any person with a military, political or economical background. In recent Android 4.0 devices, there is program named as android debugging bridge (adb) serves protection against desktop malware which serves as distribution channel for malicious apps. [4]

Frounhofer research institution [5] conclude that they developed extensible exploit execution framework which evaluate and emulate local attacks and malware. But they are not giving solution on attacks. As per our research we come on conclusion that we have to develop such security app which built in every android device. This app have root access and can block installation of malicious app. Some restrictions and ways for prevention are as follows:

1. Security App redirects each installation to registered market place like Google play store.

2. User can take access to install app from third party sites but those links get copied and sent reports to developer of security app.

3. Developer debug the app by framework and if it is suspicious then security app gives notification to user.

4. Security app have to be permitted for strictly removal of captured app.

5. Built in adb program should update by weekly from the help of Security app.

6. This security app cannot be download by any site rather than from owners sites.

7. Security app have explicit control over USB-debugging which access the internet.

REFERENCES

[1] Gartner, “Mobile Device Sales Q3 2011,” November 2011. http://www.gartner.com/it/page.jsp?id=1848514 march 16, 2015 2:24pm

[2] http://gadgets.ndtv.com/mobiles/news/windows-phone-grows-104-percent-year-over-year-in-q4-2013-abi-research-478672 march 16, 2015 2:14pm

[3] Android Team, “Platform Versions,” March 20, 2015 3:20pm. http://developer.android.com/resources/dashboard/platform-versions.html

[4] X. Jiang, “GingerMaster: First Android Malware Utilizing a Root Exploit on Android 2.3 (Gingerbread),” August 2011.

[5]http://www.cs.ncsu.edu/faculty/jiang/GingerMaster/http://www.aisec.fraunhofer.de/content/dam/aisec/Dokumente/Publikationen/Studien_TechReports/deutsch/AISEC-TR-2012-001-Android-OS-Security.pdf march 16, 2015 2:14pm

BIOGRAPHY

Vinod Shivdas Panchal born in Mumbai, Thane district, Maharashtra India. He is pursuing MCA final year in ASM's Institute of Management & Computer Studies ,IMCOST, Thane

Kanchan Sanjay Patil born in Mumbai, Thane district, Maharashtra India. She is pursuing MCA final year in ASM's Institute of Management & Computer Studies, IMCOST, Thane