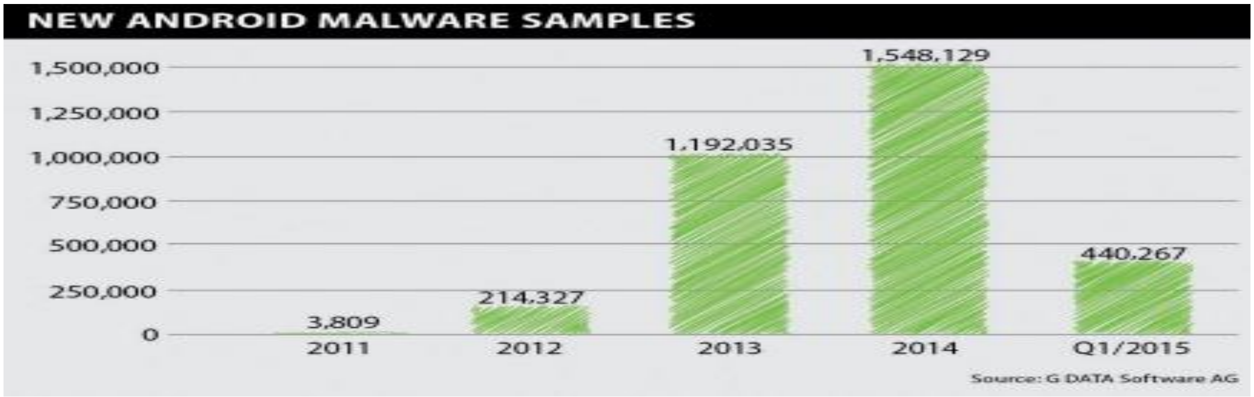
Leonard M. Pickering
Dr. Phillip Lunsford
ICTN 4040: Enterprise Information Security
6 April 2016

Current Security Issues within the Android Operating System

Android is a mobile Operating System (OS) developed by Google based on the Linux kernel. The OS was initially developed by Android, Inc. with the backing of Google in 2005. After three years of development, the first Android based mobile phone was available for sale in November of 2008. As of today, the Android operating system has been installed on over 1.4 billion devices, and is currently the most widely used computer operating system in the world. Due to the constantly increasing number of Android specific malware and other exploits, it has also become the most targeted and most vulnerable.

In recent report by the firm GData Software, approximately 440,267 new malware samples were detected in the first quarter of 2015. This number breaks down to 4,900 new threats daily or one new piece of Android Malware is released every 18 seconds. This is a 21% increase over the same period in 2014.



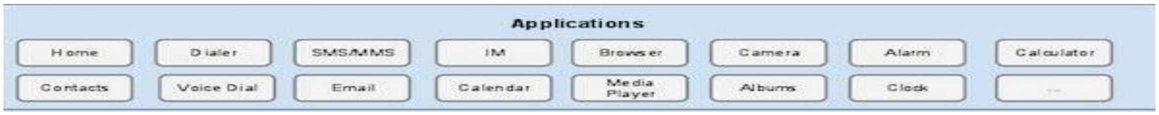
Source: G DATA Software AG

The report also specified that more than half of the new threats are financially motivated. A new twist on this type of Short Message Service (SMS) Trojan was discovered in March of 2016. This particular class of Trojan malware comes with various names, but all are essentially the same code. It is distributed from rogue websites that advertise a certain type of video content. Users are asked to download and install a special video player in order to view movies. During the installation, the malicious Android Application Package (APK) installs three other applications on the device. These then automatically use stored credit card information to subscribe to premium SMS and online video services, without the owners knowledge or consent. Among the Trojans in this class, researchers highlight Svpeng, which also comes with ransomware functionality, and Faketoken, whose purpose is to steal mobile transaction authentication numbers.

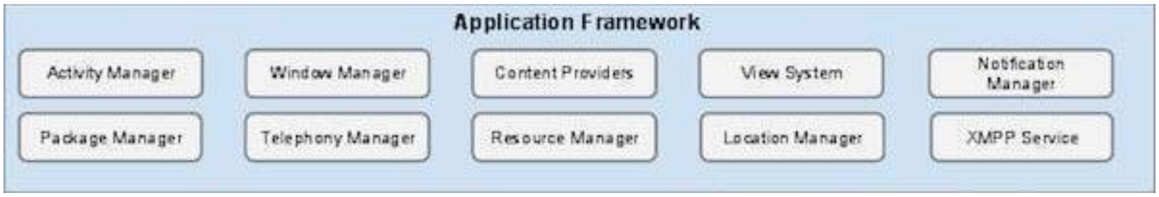
The report from GDATA states the “number (of new types of malware) could be even higher, as experts have only studied malware with a direct financial purpose. If a malware program subsequently installs apps or steals credit card data as an additional process after a payment has been made, the malware does not appear as financially motivated in these statistics.”

In order to understand current security risks within Android, we must first take a look at the Android OS architecture, and how malicious applications gain access to critical areas. Android Architecture is comprised of five basic layers: The Application Layer, the Application-Framework layer, the Library layer, the Android Runtime layer and the Linux kernel.

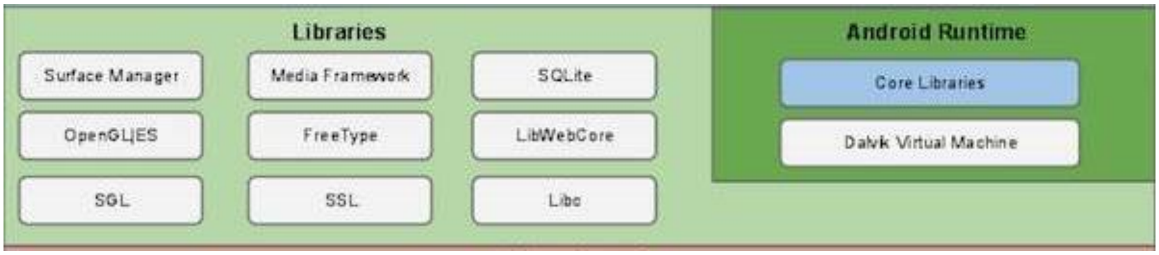
- The Application Layer is the top level layer in Android architecture. This is where the user has the most interaction and includes modules for SMS, web browsing and other user applications.



- The Application Framework layer manages basic functionality such as activity life cycle, data exchange as well as location, notification and resource management.



- The Library Layer resides on top of the Linux kernel and provides instruction sets for handling different types of data such as 2D/3D graphics, codecs for audio/video files, fonts, SQLite and OpenSSL among others.



- The Android Runtime Layer is a java based Virtual Machine (VM) specifically designed and optimized for Android. This allows each application to run in its own process and have its own instance of a Dalvik VM.
- The Linux Kernel is updated in relation to the version of Android OS. The current version, 6.0 Marshmallow uses 3.18.10 while Android 4.4 KitKat uses 3.10. Users and developers do not directly interact with the kernel, but is the core of the whole system. The kernel provides basic system functionality, such as device, memory and process management, as well as providing drivers for attached peripherals. When you boot an

> Android device, the Linux kernel loads just like it would on a Linux distribution, but, much of the other software is different. Android doesn't include the GNU C Library (glibc) used on standard Linux distributions, or all of the other GNU libraries you would find on a typical Linux distribution. It also doesn't include an X server like Xorg, so it is impossible to run standard graphical Linux applications in an Android environment.

The Android operating system uses permissions in order to allow or prohibit applications and components to interact with other applications/components or critical resources. User approval is required before getting access to critical operations such as telephone operations, SMS and GPS within in an application. Applications then request the permissions in order to execute successfully and declare the permissions in its configuration file (AnroidManifiest.xml).

When an application is installed, android prompts the user to either allow or reject these requested permissions. Once installed however, an application has no further obligation to inform the user of how, when or why it is accessing other applications it was granted permission to, and the only discernable way to stop it, is to uninstall the app.

This type of functional vulnerability surfaced in 2015 with what is known as "Stagefright" hack. The scope of this paper is too narrow to explain exactly how the Stagefright hack works. In short, the built in SMS application, Hangouts, had inherent permissions to the built in Gallery app. Videos sent via SMS are processed automatically without even opening them. Using a modified video file, a hacker could inject malicious code into any of over 950 million Android devices without the user being aware. Once the video is received, any one of a number of exploits could be deployed allowing complete takeover of the user's device. When discovered, this exploit opened up more inherent flaws though not within the Android OS, but within the

Android ecosystem and the Original Equipment Manufacturer (OEM) supply chain and Google itself.

The open source nature of Android allows a company to adapt its code for use in their hardware. Besides phones and tablets, Android OS is used in home appliances, cars and even toilets. Google has adopted a 180 day rollout schedule for updates to Android. This often leaves consumers with outdated software on Android smartphones. When a new version of Android is released, OEM's also have to port their custom software over to the latest versions. This results in OS version fragmentation. Android is embedded OS. There are no generic device drivers. Manufacturer updates are custom to each model they produce. With new and inexpensive hardware being continually released, it is not fiscally sound to roll out complete custom OS updates when devices today have a two year life cycle and low profit margins. If an update is pushed, older hardware capability limitations can render a certain features nonfunctional or the device itself inoperable. OEM Manufacturers are also bound to a myriad of cell phone carriers, who, must certify and push the update. If they opt to deploy OS updates furnished by manufacturers, the delay can be months or even years.

The Android platform is criticized for the time it takes to get critical updates to consumers. Google is only responsible for the Android Open Source Project (ASOP) which is technically just the operating system. Unlike Android OS updates which come on new devices, Google, OEM's and carriers have been slow to patch security issues in older releases. With Stagefright, Google had a patch ready but getting it through the ecosystem was extremely difficult. Recently, Google has dropped support for Android versions 4.3 and below. This has left manufacturers responsible for the development and implementation of security updates, and the mobile phone carriers to individually push them. Which essentially will not happen. This has effectively left over 95% of android users with a vulnerable operating system. Specifically in the Stagefright case meant 5.1 Lollipop and above were the only Android OS patched, accounting for only 4-5% of total devices. This is the equivalent of Microsoft relying on Dell or Hewlett Packard and Lenovo to develop and implement security patches for Windows 8.1 while also cutting off all support for Windows 7 and below.

The publicity surrounding the original Stagefright has led to an overhaul in the way that updates were handled by Google, manufacturers and phone carriers. On September 10th, 2015 Google started to push monthly security updates for its Nexus device owners. The first patch included a security fix that allowed applications to bypass the SMS notifications prompting users for permission to access other apps. Many other manufactures, prompted by public awareness of less than secure operating systems revamped their security policies. Samsung has taken an aggressive stance and has monthly updates as well.

The outlook for Android is looking good. Starting in 2010, Google started to deconstruct the Android OS. What this entailed was pulling out embedded applications that were considered non-essential to the actual OS and placing them as separate apps in the Google Play store. The first to go was G-mail. What followed was a major list of applications and services which were then moved into smaller standalone positions within the Android structure. In March of 2016, the last to app to go, Google Calculator, was officially released as a standalone app. This strategy of less internal bloatware allows OEM's to make OS level modifications more easily. It also allows Google to update key components not associated with the actual OS in real time while giving consumers a choice of not using Google's in house apps for key functions.

The latest preview, Android N, suggests that Google is experimenting with separating the kernel from all surface level services. Googles has begun to divide the Android OS into two distinct sections. The core OS which includes the kernel and framework, and the user interface. This would give OEM’s more control over the outer core of the Android experience while not facilitating system wide changes that affect and inhibit the ability to update core functions. This is a direct reference to security updates. If and when this Android OS model is fully deployed system updates would become more streamlined and would fix the Android operating systems biggest problem: eliminating OEM’s and telephone carriers from the native Android OS bug repair business and placing it directly on Google’s shoulders.

Current best Android security practices include two items of note to secure your OS. Google recommends you only download and install apps from the Google Play Store. Prior to being available on the Play Store, apps go through extensive scrutiny to ensure developers adhere to all of its policies. Should Google find apps that violate its policies, these then will be blocked from the app store and the developers may be banned. To keep Android devices as safe as possible, Google continues monitoring even after the apps have been certified.

The second concerns the apps you install on your Android device. They should not be able to access other stored data, such as your location or photos or financial information unless you grant them permission. If you own a device running on Android Marshmallow, you can manage an app's permissions by heading to Settings, Apps and choosing the app from the list of installed apps. The list of all the permissions that the app has access to should now be ready to edit. You can then toggle permissions as you see fit. If you happen to own a device with an older version of Android, app permissions are set during install and cannot be changed. Best practice states to uninstall then reinstall the app, changing permissions as you see fit.

The Android OS has exploded into our lives in less than ten years. Within the last twelve months, The Android operating system has been found to be susceptible to quite a few attacks through numerous vulnerabilities. Security patches and updates are a normal occurrence for every operating system available, but none are as hindered by the android ecosystem that has been established. Though not officially announced, it appears that Google is taking the appropriate steps to rectify this situation. The problem is that we will all have to purchase new devices to take full advantage of a more secure and patchable OS.

References

Muneer Ahmad Dar, Javed Parvez, “*Evaluating Smartphone Application Security: A Case Study on Android”*, Global Journal of Computer Science and Technology Network, Web & Security, 2013, Volume 13, Issue 2 * https://globaljournals.org/GJCST_Volume13/2-Evaluating-Smartphone-Application.pdf

Metagar S. M., Motupalli J., Theja N., Bhootaleppa B. P.,”*Mobile Security in Android Mobile Technology”*, International Journal of Research In Information Technology, February 2013, Volume 1, Issue 2, Pg. 30-36 * https://www.academia.edu/2943362/MOBILE_SECURITY_IN_ANDROID_MOBILE_TECHNOLOGY

Waqas A., “*Hackers Develop Android Malware every 17 seconds”* Hackread, 14 July 2015 https://www.hackread.com/android-malware-development-17-sec/

Abelson A. “*Introduction to android Development”* IBM Development Works, 12 May 2009, http://www.ibm.com/developerworks/library/os-android-devel/

Siddiqui A., *“Stagefright Explained: The Exploit that Changed Android”* XDA Developers, 14 August 2015 http://www.xda-developers.com/stagefright-explained-the-exploit-that-changed-android/

Hill S. *“10 Crazy Android Devices That Makes Your Phone Look Boring, From Fridges to Bikes”* Digital Trends, 21 August 2014 http://www.digitaltrends.com/mobile/10-crazy-devices-that-have-run-android/

Brandom R. “*How the Stagefright Bug Changed Android Security”* The Verge, 5 August 2015 http://www.theverge.com/2015/8/5/9099627/google-stagefright-android-vulnerability-protect-patch

Raphael J.R. *“What one silly app reveals about Google’s grand Plan for Android”* Computer World, 31 March 2016 http://www.computerworld.com/article/3049200/android/google-grand-plan-android.html

Smith C. *Surprise: It turns out there is a potentially huge security flaw in Androids app store”* BGR, 20 June 2014 http://bgr.com/2014/06/20/google-play-android-apps-security-flaw/

Biswas S., Haipeng W., Rashid J., *“Android Permissions Management at App Installing”* International Journal of Security and Its Applications, 2016 Volume 10, Issue 3 * http://www.sersc.org/journals/IJSIA/vol10_no3_2016/21.pdf

Park J., Choi S., "*Studying Security Weaknesses of Android System"* International Journal of Security and Its Applications, 2015 Volume 9 Issue 3* http://www.sersc.org/journals/IJSIA/vol9_no3_2015/2.pdf

Amadeo R., *"Waiting for Android's Inevitable Security Armageddon"* ARS Technica, 6 August 2015 http://arstechnica.com/gadgets/2015/08/waiting-for-androids-inevitable-security-armageddon/

Amadeo R. *First-ever monthly Android Security updates start to roll out"* ARS Technica, 10 September 2015 http://arstechnica.com/gadgets/2015/09/first-ever-monthly-android-security-updates-start-to-roll-out/

Amadeo R. *"The state of Android updates: Who's fast, who's slow, and why: Naming and shaming (sometimes praising) the update efforts of OEMs and carriers"* ARS Technica, 20 August 2014 http://arstechnica.com/gadgets/2014/08/the-state-of-android-updates-whos-fast-whos-slow-and-why/3/

Brandon R., *"How the Stagefright bug changed Android security"* The Verge, 5 August 2015 http://www.theverge.com/2015/8/5/9099627/google-stagefright-android-vulnerability-protect-patch

G Data, (2015) "*Mobile Malware Report*" [White paper]. Retrieved 31 March 2015, from G Data Software https://public.gdatasoftware.com/Presse/Publikationen/Malware_Reports/G_DATA_MobileMWR_Q1_2015_US.pdf

Carlon K., *"Why monthly security updates won't make your phone any safer"* Android Pit, 6 October 2015 https://www.androidpit.com/android-monthly-security-updates

Swati K. *"How to Hack Millions of Android Phones Using Stagefright Bug, Without Sending MMS"* The Hacker News, 31 July 2015 http://thehackernews.com/2015/07/how-to-hack-android-phone.html

Leake G., Madenci L. *"Android Architecture & Boot Process"* GitHub, 14 February 2014, https://github.com/lmadenci/Android-Architecture-Overview