**Shashank Gowdagiri**

**1956-1095-02**

**Investigation of the Android OS: The rise and rise of Malware and Countermeasures**

**CSci530 Computer Security Systems**

I have read the Guide to Avoiding Plagiarism published by the student affairs office. I understand what is expected of me with respect to properly citing sources, and how to avoid representing the work of others as my own. The material in this paper was written by me, except for such material that is quoted or indented and properly cited to indicated the sources of the material. I understand that using the words of others, and simply tagging the sentence, paragraph, or section with a tag to the copied source does not constitute proper citation and that if such materiel is used verbatim or paraphrased it must be specifically conveyed (such as through the use of quotation marks or indentation) together with the citation. I further understand that overuse of properly cited quotations to avoid conveying the information in my own words, while it will not subject me to disciplinary action, does convey to the instructor that I do not understand the material enough to explain it in my own words, and will likely result in a lesser grade on the paper.

Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Shashank Gowdagiri\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Investigation of the Android OS: The rise and rise of Malware and Countermeasures**

Shashank Gowdagiri

Department of Computer Science

University of Southern California

sgowdagi@usc.edu

**Abstract- The smartphone has gone a long way, from being an innovative idea barely 5 years ago to becoming a necessity today. There is an ever growing presence of smartphones in our daily lives, however with the usage of advancing technology, comes the inevitable problem of more threats and, in this case, Malware. There is a need to curb these intrusions and make safe the information of the user provided to the device.**

**In this paper, the Android operating system is analyzed in depth, contrasted with another popular operating system, the iOS and the security measures taken by each of them to protect the users against different types of malware and the extent of their success. Further, there is also a section devoted to the different origins of malware, with emphasis on the ways of intrusion and detection mechanisms. The paper additionally describes a series of approaches designed to safeguard the user’s information and traces the security pitfalls of each of the releases of the Android operating system.**

**Keywords: Smartphone Security, Android, iOS**

**I .Introduction**

There is no denying of the recent explosion of the smartphone industry in recent years. The first Android powered smartphone was released in October 2008, and it has grown to 1 billion active smartphones in 2013, with around 48 billion apps being downloaded [1]. The primary reason for the unprecedented growth of the Android OS is twofold. First, the source code of the operating system is open source [2] thereby giving access to developers and enthusiasts to freely modify the source code and build their own software. The second reason is the unrestricted publishing of “apps” - (a colloquial term for applications which is built by companies or individuals) on the Google Play store, and the option to install apps through an unknown source [3] .This inherently makes a hacker’s playground more joyful. The painful process of allowing selective and vastly lesser number of applications to be published on the App store in contrast – which includes validation, code changes [4] and verification of identity is much more efficient. Developers however, prefer the Julian Assange route and support for more freedom for distributing their code and apps through a series of more capitalistic payoff arguments. This has led to an astounding 71% of developers sticking to Android’s Google play and releasing their apps there (there is also unrestricted Java Program usage which is wildly popular) as opposed to iOS’ App Store [5]. Of course, this doesn’t mean that the iOS is completely secure- a prime example came in March 2010 when Becher, Freiling and Hoffman point out the entire SMS database being stolen by Iozzo and Weinmann [6] by creating a drive-by download app against an iPhone 3GS [7] . An advanced understanding of an operating system is expected to fully understand the Android and the iOS architecture, which will be explained more in depth in the next section, entitled Android System Architecture. The security model for the Android OS is scrutinized further in the same section. The main reason as to why hackers prefer attacking smartphones as explained by Richard Mulliner [Mulliner 2006]:

* **Mobility**: This is the most important characteristic of the mobile phones. Since mobile users can take them to anywhere, the chances of getting stolen, lost, or physically tempered increases as compared to stationary devices.
* **Strong Personalization**: As a personal device, mobile devices usually are not shared among multiple users.
* Strong Connectivity: Mobile phones are commonly used to connect to other devices over the wireless networks (or wireless Internet) for data exchanges.
* **Technology Convergence**: Today numerous functional features are integrated in the mobile phones, for example gaming, video and data sharing, and internet browsing.
* **Limited Resources and Reduced Capabilities**: Comparing with stationary devices, mobile devices have four major limitations: a) limited battery life, b) limited computing power, c) very small display screen size, and d) very small sized keys for inputs. These limits bring the challenges in building mobile security technology.

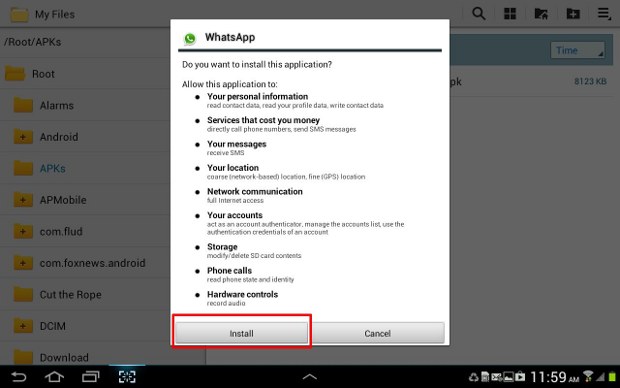
Describing the type of threats that each of the malware types by Yajin Zhou and Xuxian Jiang [8] would make the countermeasures against them almost impossible to solve. There are more known threats today than there were a year ago by a factor of 100. The section entitled Malware Deconstructed gives more depth to this.

**Android System Architecture**

The basic rules of an operating system apply. It is built on a Linux Kernel, for mobile devices such as smartphones and tablet computers, led by Google. It is essentially a sandbox. The Android OS was built with a view to make mobile devices more useful and give the user more freedom and more capability at the touch of their fingertips. Unfortunately, with more power of the device, the easier it has become to penetrate such technology. The android security model is explained clearly below [9]:



The starting shell scripts to power up the device and the basic framework for the Android library is explained more in detail by Google, since it is open source. However, a snapshot of the working kernel and the executables that are present on the device is broken up. [10] A detailed overview of the security model is presented in various other papers. However, the thorough summarization of the entire architecture was based on a single primary security model – with a view of privileges and separation. The essential idea for any application to run on the android device is that it needs user permission to access any underlying system resources. A common example would be the installation of a location based service – something like Whatsapp. The most common installation screen that a user sees would be permission to access location, SMS details, phone contacts, Facebook profile, and Gmail account settings (which comes in by default owing to the Google technology).



Here in lies the danger of providing access to applications which are installed from unknown sources. Sometimes applications use private information stored on the system or of other application space. [11] The first layer of protection offered by Android is the remote uninstallation capability from other applications like Gtalk or Gmail, which target applications known to have viruses. This is done on the Google play market, where users might have given a negative rating, the source code might be outdated, or previous users have had a terrible experience with it. It is part of the Google initiative to cut down on malicious applications. [12, 13] Moving to the security model analysis of the Android OS, it has greatly been beefed up since the inception of Android.

According to recent Computer blogs and the Wikipedia Android Mobile Security content, [17] Android smartphones have the ability to report the location of Wi-Fi access points, encountered as phone users move around, to build databases containing the physical locations of hundreds of millions of such access points. These databases form electronic maps to locate smartphones, allowing them to run apps like Foursquare 7.0 [14], Google Latitude [15] and Facebook Places. Recent Development include the much In August 2013, Google released the Android Device Manager, a component that allows users to remotely track, locate, and wipe their Android device through an online interface. As it is implemented through Google Play Services instead of within Android itself, it is available to most Android devices with version 2.2 and higher. [18]

Another common way of attacking the smartphone user, which is becoming increasingly popular, is using the API to make calls to paid services, thereby resulting in the owner of smartphone to incur charges. More common attacks would include the theft of personal data, like photos, videos, and give a third interested party that data. Further, obtaining access to the point of entry of the phone and making the phone unusable is a common way of mobile security consequence. With that, we dive deeper into types of attacks and the root causes and mechanisms by which they propagate [19]

**Malware Deconstructed and Attacks**

There are different types of mobile threats. However, this paper will concentrate on the software centric attacks and concentrate on the Malware which is installed, and the general pattern followed by the Malware to activate itself into the system. After this, a couple of countermeasures of mitigating software solutions are provided.

As pointed out by Becher, Hoffman, Holz, Uellenbeck and Wolf [19], we can categorize the attacks into hardware centric, software centric, device independent attacks and user layer attacks. Since we are concentrating on Malware, the following section will bring some aspects about software-centric attacks first, before delving into Malware.

Software centric attacks can be further categorized into six distinct components, with an additional overview given about each of them [18, 19]

First, the impact of Malware on the system, is it taking personal data, is it charging the user through calls to paid services via hacking, or is it performing some sort of identity theft, eavesdropping etc. Stated by Becher and Co, [19] One example would be the *Trojan-SMS.AndroidOS.FakePlayer*, which

Pretends to be a movie player, but secretly sends messages to a service number which are highly charged [20]. The ideal of a mobile botnet – which is unknowingly running as a remote controlled machine at the fingertips of the attacker would make a very compromising device seem harmless if the user does not know that it exists.

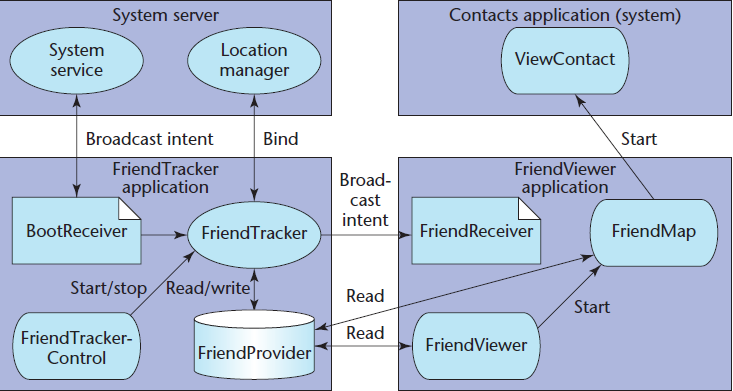
The next two types of attacks on the software centric attacks based Malware infected mobile system are similar in nature – the SMS and the MMS . However, the SMS based attacks are reducing in number and are posing to be lesser of a risk as the usage of smartphones goes up. The mail threat now is the data downloaded over an unsecure Wi-Fi channel.[20]

Similar to the way viruses and other undesirable programs are unknowingly downloaded from the internet onto a laptop using a browser, the mobile browser also is a channel for malware to find itself on the phone. The ways of entry into the phone is described in the next section when Malware is discussed, and its process of activation and harm. However, with recent standards being enforced on the web browser on a smartphone, all calls made via a hyper link must have explicit permission from the user. Examples would include the jailbreak of an iPhone [21]

Malware which is downloaded onto the physical device makes tracking and removal harder due to limited processing power and lesser ease of access to system resources.[18] However, we do have a number of ways to detect Malware – Signature based detection, anomaly based detection are two classic techniques. According to Portokalidis et al. [22] we can also use a “tracer” program.

The last component under which software centric attacks can be spoken about are the approaches the buffing up the operating system itself, with techniques like kernel hardening, stingy permission rights and process isolation etc. This causes a tradeoff for the security for the phone with a somewhat less powerful one.

A brief overview about the application’s interaction with the Android OS is given below [23]:



**Keyword: ADB: Android Development Bridge**

**Malware**

Installation: Malware can spread from many sources. Some of the most common ways of infecting a system is through repackaging, where the authors locate the source at which any popular application is released or being “patched”, and they introduce their malicious code there. [11]

***“For example, AnserverBot malware uses a package name com.sec.android.provider.drm for its payload, which looks like a module that provides legitimate DRM functionality”***

Another very popular way of spreading this potentially harmful software is running the malicious code at run time or when the popular application is being updated. [11] The following snapshot has been taken from a research paper entitled “Dissecting Android Malware”



There are other ways, such as drive-by download, but this paper will not concentrate much on that.

The Malware then activates itself on the system, sometimes through simple procedures like a call. [30] The application received higher privileges, since it is piggybacking on another app, called privilege escalation. Of course, the malware could be targeted to obtain personal information, or an identity theft or for a variety of other reasons.

The iOS has been famed to be impenetrable, but there is evidence of the first discovered malware on the iOS platform. This is from the Wikipedia Mobile Malware webpage[31]

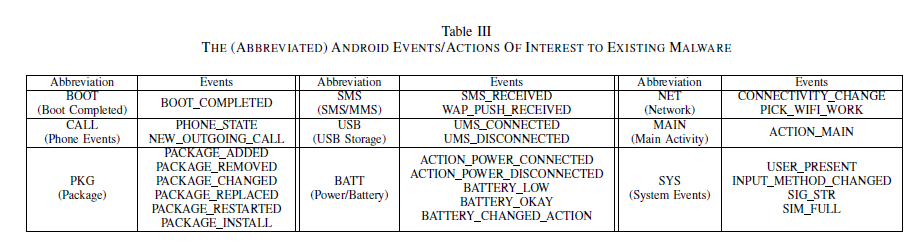
* ***“Ikee: The first worm known for iOS platforms. It only works on terminals that were previously made a process of jailbreak, and spreads by trying to access other devices using the SSHprotocol, first through the subnet that is connected to the device. Then, it repeats the process generating a random range and finally uses some preset ranges corresponding to the IP addressof certain telephone companies. Once the computer is infected, the wallpaper is replaced by a photograph of the singer Rick Astley of RickRoll.”***

There is a growing trend, when Android replaced Symbian OS as the most popular mobile operating system on the planet, there were about 100 or so cases on Symbian at that time with mild to minimal consequences.

However, as Android became more popular, the number of malicious software releases has grown immensely in a very short period of time. This makes for one of the best ages to be a mobile security developer, as well as a hacker, since everything is open source.

**Countermeasures and Mitigations**

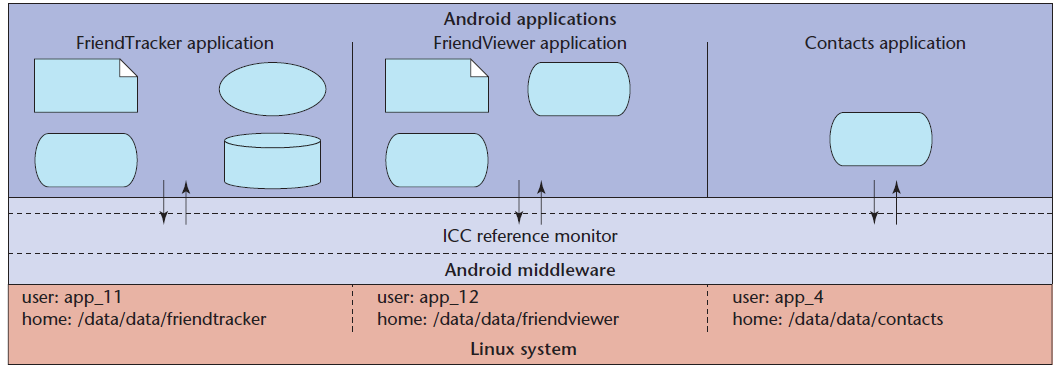
There are many different types of malware, and a brief table is given to describe some characteristics about them [11]



The most influential and perhaps popular kinds of malware that exist are the DroidKungFu [24] and the AnserverBot. [25]

Looking ahead to the countermeasures that the Android OS takes to ensure it security, it seems that the Android OS is remarkably similar to the iOS standards. It also employs the same features as any operating system, as described below, and there are some suggestions on the improvements.

First, let us take a look at the security provided at a basic level by the Android OS [23]



Android “patches” or software fixes have never been a new concept, but the remote installation of a new software patch on applications installed (when the default settings are to update automatically) causes a big problem. As we know, the AnserverBot Trojan bot piggybacks on legitimate applications, and then activates itself and causes itself to multiply and encroach on the memory space of other applications. This in turn, makes the memory get unnecessarily fully utilized for the entire phone, and this causes delays in the OS, hence causing all the processes to eventually get into a deadlock, and then crash the phone for good. This has become especially common when the Wi-Fi that the smartphone device is currently connected to is an unsecure one, and there would be many software “patches” that are helping in carrying these threats directly to the users’ phones.

The first level of security as a user would be to ensure strong passwords, a design pattern, or a combination of both to access the most sensitive apps which require large amounts of memory. As mentioned above, there are quite a lot of users who agree to install apps from unknown sources. This causes the updates to occur from an unknown source.

There are quite a few tips and tricks which is based on the model of the Android OS itself, which gives the users many rights there didn’t know existed. Some of them include:

* Making use of the permissions model which Android has become famous for. Running applications with higher privileges and setting custom permissions for sensitive applications
* There is an experimental new concept, in which real time data about how the application interfaces with the system resources and what is exactly happening can be seen by the user called TaintDroid [26]. An excerpt from that paper states the following use of TaintDroid:

*“Using TaintDroid to monitor the behavior of*

*30 popular third-party Android applications, we found 68 instances of potential misuse of users’ private information across 20 applications. Monitoring sensitive data with TaintDroid provides informed use of third-party applications*

*for phone users and valuable input for smartphone security service firms seeking to identify misbehaving applications”*

* A major concern is also an authenticated Android Development Bridge. [27] . Vidas ,Votipka and Christin argue that the authorized user must be able to login anytime he wants to authenticate the ADB once the connection is made. [11]
* Certain specific examples can give reason to how Android Is actually quite strong by itself but some applications do “weird” things by themselves. The following is an excerpt from a recent tech blog:

*“In the general case, each application*

*runs as a unique user*

*identity, which lets Android limit*

*the potential damage of programming*

*flaws. For example, the Web*

*browser vulnerability discovered*

*recently after the official release of*

*T-Mobile G1 phones only affected*

*the We b browser itself (http://*

*security evaluators.com/content/*

*case-studies/android/index.jsp).*

*Because of this design choice, the*

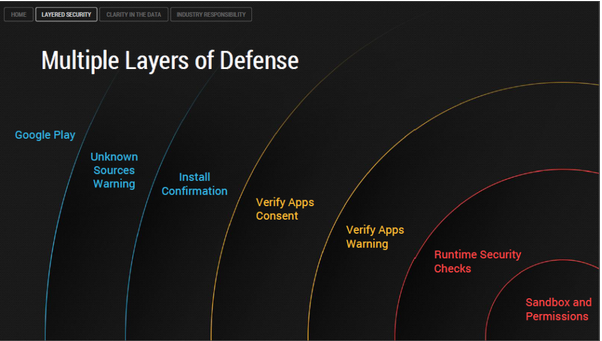
*exploit couldn’t affect other applications*

*or the system.”*

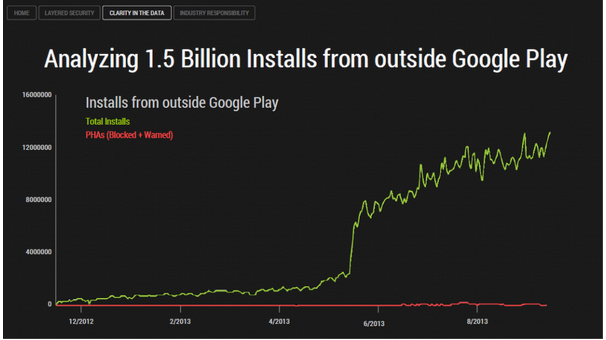
This shows that the system is essentially built to protect the user at most costs, but certain applications are just built in a unique way.

* Some attackers try to change the boot image, so the obvious use of a TPM system to ensure that the device is authenticated, and provide a ground truth from which to “build up”[11]

There are however, some that say that the Android is impenetrable. Reaching all the way to the kernel to modify the root image would be an impossible task. The kernel has been sandboxed very well is Google’s claim. The unofficial representation of the actual inner workings of the Android OS is shows below [28]



To illustrate this:



To immediately counter this logic, there was a leaked paper which claimed that a text message brought malware into the Android OS and they were proved true. According to a now-leaked and unofficial document [29] a solitary text message could do such harm.

**Conclusion**

A series of events that have led to people with malicious intent to crack down and penetrate one of the most revolutionary and innovative systems of all time is where we stand now. The malware that is being propagated is relentless, and the Android development team can only hold the fort down for so long. It eventually becomes inevitable, that the Android and iOS teams sit down, and hash out a plan to avoid all kinds of attacks, by strengthening their entire system and structure. A further, depth intensive coverage of this topic would reveal that there are many more flaws than the ones presented as an overview in this paper, but with every coming day, there is a threat that somebody is waiting to release. It is logistically impossible to trace the person behind every Trojan or Bot or Malware propagator, but all that the developers building the operating systems can do, is build something unbreakable. It is time for the Trojan bugs to stop penetrating through the high walls of the kingdom of Android in Troy. The next generation of anti-malware experts and the Google Play developers need to raise their game to a whole new level, because as it stands right now, the number of threats outweigh the protection that is offered to fend them off.

**Acknowledgements**

I would like to thank the course instructor Dr. Clifford Neuman for his support and kindness. His lectures have been interesting and have helped me write this paper from scratch. I would also like to thank my fellow students and the University of Southern California for its resources which I have used immeasurably to complete this paper.

**References**

[1]Vic Gundotra Blog <https://plus.google.com/+VicGundotra/posts/8CVJ79nPQwN>

[2] Android Source Code download <http://source.android.com/source/downloading.html#installing-repo>

[3] Android Developer Publishing Guideline <http://developer.android.com/distribute/googleplay/publish/preparing.html>

[4]How to submit an App to the App Store <http://www.idev101.com/code/Distribution/app_submit.html>

[5] Economics of developing Apps <http://www.developereconomics.com/reports/q3-2013/>

[6] A. Portnoy, “Pwn2Own 2010,” 2010, <http://dvlabs>.

Tippingpoint.com/blog/2010/02/15/pwn2own-2010.

[7] Mobile Security Catching Up?

Michael Becher, Felix C. Freiling

University of Mannheim, Germany

Johannes Hoffmann, Thorsten Holz, Sebastian Uellenbeck, Christopher Wolf

Horst G¨ortz Institute (HGI)

Ruhr-University Bochum, Germany

[8] Android Security Model <http://www.cs.ncsu.edu/faculty/jiang/pubs/CCS13.pdf>

[9] Android building blocks <http://elinux.org/Android_Architecture>

[10] Android – Under the hood <http://benno.id.au/blog/2007/11/13/android-under-the-hood>

[11] All Your Droid Are Belong To Us: A Survey of Current Android Attacks

Timothy Vidas

ECE/CyLab

Carnegie Mellon University

Daniel Votipka

INI/CyLab

Carnegie Mellon University

Nicolas Christin

INI/CyLab

Carnegie Mellon University

[12] <http://www.theregister.co.uk/2013/10/21/android_ios_security_comparison/>

[13] <http://arstechnica.com/security/2013/04/family-of-badnews-malware-in-google-play-downloaded-up-to-9-million-times/>

[14] TechCrunch Article <http://techcrunch.com/2013/12/05/foursquare-7-0-brings-scanable-design-and-expands-proactive-recommendations-to-all-users/>

[15] <http://en.wikipedia.org/wiki/Google_Latitude>

[16] <http://en.wikipedia.org/wiki/Mobile_security>

[17] Android Antivirus review <http://www.zdnet.com/android-antivirus-comparison-review-malware-symantec-mcafee-kaspersky-sophos-norton-7000019189/>

[18] Google play support <https://support.google.com/googleplay/answer/2479847?hl=en>

[19] Mobile Security Catching Up?

Michael Becher, Felix C. Freiling

University of Mannheim, Germany

Johannes Hoffmann, Thorsten Holz, Sebastian Uellenbeck, Christopher Wolf

Horst G¨ortz Institute (HGI)

Ruhr-University Bochum, Germany

[20] Kaspersky Lab, “First SMS Trojan Detected for Smartphones

running Android,” 2010, ”http://www.kaspersky.

Com/news?id=207576156”.

[21] Jailbreak for iPhone <http://www.jailbreaknation.com/jailbreak-6.1-untethered-with-evasi0n-for-ios-6-iphone-5-4s-4-3gs-ipod-touch-5g-4g-ipad-mini-432>

[22] G. Portokalidis et al., “Paranoid Android: Versatile Protection

For Smartphones,” in Annual Computer Security

Applications Conference (ACSAC), 2010

[23] **Dissecting Android Malware: Characterization and Evolution**

Yajin Zhou

*Department of Computer Science*

*North Carolina State University*

*yajin* [*zhou@ncsu.edu*](mailto:zhou@ncsu.edu)

Xuxian Jiang

*Department of Computer Science*

*North Carolina State University*

[*jiang@cs.ncsu.edu*](mailto:jiang@cs.ncsu.edu)

[24] DroidKungFu is back again! <http://www.csc.ncsu.edu/faculty/jiang/DroidKungFu3/>

[25] An analysis of the AnserverBot Trojan <http://www.csc.ncsu.edu/faculty/jiang/pubs/AnserverBot_Analysis.pdf>

[26] TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones

William Enck, Peter Gilbert, Byung-Gon Chun Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, Anmol N. Sheth

<https://www.usenix.org/legacy/event/osdi10/tech/full_papers/Enck.pdf>

[27] Android ADB <http://developer.android.com/tools/help/adb.html>

[28] Android impenetrable?

<http://qz.com/131436/contrary-to-what-youve-heard-android-is-almost-impenetrable-to-malware/>

[29] :Leaked report – Text message malware on Android

<http://info.publicintelligence.net/DHS-FBI-AndroidThreats.pdf>

**[**30] Android Malware Case

<http://www.h-online.com/open/news/item/Android-malware-activates-itself-through-incoming-calls-1253807.html>

[31] Mobile Malware - IKEE

<http://en.wikipedia.org/wiki/Mobile_malware>