International Journal of Electronic and Electrical Engineering.
ISSN 0974-2174 Volume 7, Number 7 (2014), pp. 717-722

http://www.irphouse.com

Literature Analysis on Malware Detection

Parmjit Kaur[1] and Sumit Sharma[2]

[1]Department of CSE, PG Student,
CGC Group of Collages, Gharuan, Mohali, India
[2]Department of CSE, Chandigarh University, Gharuan, Mohali, India

Abstract

Usage of Android smartphones is more as compared to another smartphones due to its Open Source Operating System. Due to its Open OS, Android enables us to install third party applications. However, Security is one of the main concerns in Android. Security threats of malicious applications are rapidly increasing due to the nature of the third party applications where only developers can assign required permissions. Applications are installed on the basis of all or nothing basis. For this reason, attackers can inject into a normal application with inappropriately acquired permissions. In this paper, we have described the android architecture, various types of malware and literature analysis for security considerations in android smartphones, including the various general approaches and techniques for detection of various malwares.

Keywords: Malware, Smartphone, Android, Operating System

1. Introduction:

In these days, usage of smartphone becomes steadily increased. Every person wants smart phones like Android, iPhone, Symbian but usage of Android smartphones is more as compared to another smartphones due to its Open Source Operating System [1]. Open source defines the source code for any application which is being developed not for sale or profit oriented purposes. Without any license fee, Users and developers use the source code, but they want to go along with the rule and condition in the license's terms and conditions [5]. Due to its Open OS, Android, enables us to install third party applications. Security is one of the main concerns in Android. However, security threats of malicious applications are rapidly increasing due to the nature of the third party applications where only developers can assign required permissions. In android application, there is a list of permissions defined with <uses Permission> tag in Android. Manifest. XML file. For this reason, attackers can inject exploits into a normal application with inappropriately acquired permissions. In this paper, we have

described about the android architecture, various types of malware and the literature analysis for security considerations in android smartphones, including the various general approaches and techniques for detection of various malwares.

1. 1 Android Architecture:-

Android is composed of four layers, as can be seen in fig. 1:

- **Application:** This is the layer for the application installed (i. e. phone, mail, etc)
- **Application Framework:** provide different packages of service applications
- **Android Runtime and Libraries:** contains a core component called the Dalvik virtual machine and each process is executed in a separated instance in the VM. Also, in this layer, there are some libraries like SSL, SQLite or libc
- **Linux Kernel:** it abstracts the hardware from the software [3]

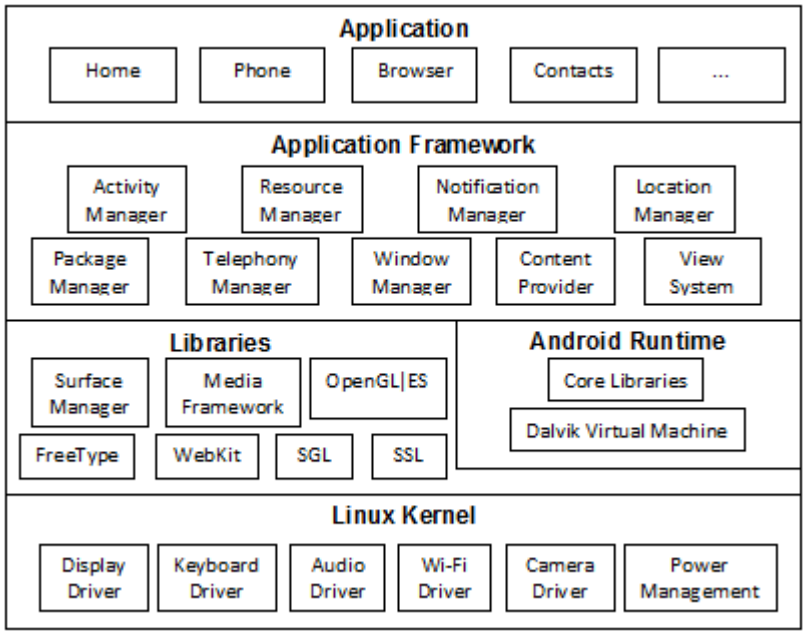

Fig. 1 Android Architecture[1]

1. 2 What is Malware:

Software that "*deliberately fulfills the harmful intent of an attacker*" is commonly referred to as malicious software or malware [Moser et al. 2007a]. Terms, such as "worm", "virus", or "Trojan horse" are used to classify malware samples that exhibit similar malicious behavior. The first instances of malicious software were viruses [2].

Types of Malware

This section gives a brief overview of the different classes of malware programs that have been observed in the wild.

A. Viruses: Computer virus refers to a small program with harmful intent and has ability to replicate self. When file is run, virus code gets executed. A virus may spread from an infected computer to other through network or corrupted media such as floppy disks, USB drives [4].

B. Worms: Worms are self replicating programs. It uses network to send copies of itself to other systems invisibly without user authorization. Worms may cause harm to network by consuming the bandwidth. Unlike virus the worms do not need the support of any file. It might delete files, encrypt files in as crypto viral extortion attack or send junk email. Example Sasser, My Doom, Blaster, Melissa etc [4].

C. Spyware: Spyware is a collective term for software which monitors an gathers personal information about the user like the pages frequently visited, email address, credit card number, key pressed by user etc. It generally enters a system when free or trial software is downloaded [4].

D. Adware: Adware or advertising-supported software automatically plays, displays, or downloads advertisements to a computer after malicious software is installed or application is used. This piece of code is generally embedded into free software. The most common adware programs are free games, peer-to-peer clients like KaZaa, BearShare etc [4].

E. Trojans: Trojan horses emulate behavior of an authentic program such as login shell and hijacks user password to gain control of system remotely. Other malicious activities may include monitoring of system, damages system resources such as files or disk data, denies specific services [4].

1. 3 Security Mechanism:

In 2012, A_PINTO [7] gives a report *Android Malware 400% increase* in which he has described how Android Malware increases rapidly. A recent study conducted by a unique organization dedicated to conduct security vulnerability research found that there is a 400% increase in Android malware since Summer 2010. Malware is a piece of software created to stealthy operate behind the user interface but that can gather sensitive information that can be used for many different purposes, going from targeted advertisement, tracking purposes up to fraud activity leading in many cases customers with unrecoverable bills. The main purpose of this post is not to make the reader aware of the existence of Malware for mobile devices, but the main purpose is to make the reader aware of the impressive grow of malware being targeted to mobile devices. And to make the reader aware of the upcoming phase of hacking techniques as Smartphones are becoming more popular, hackers have evolve from targeting PC to targeting smartphones, and particularly the Google Android OS.

In 2010, *Blasing et. al.* [6] presented a paper in IEEE named *An Android Application Sandbox System for Suspicious Software Detection.* This paper describes An Android Application Sandbox (AASandbox) which is able to perform both static and dynamic analysis on Android programs to automatically detect suspicious applications. Static analysis scans the software for malicious patterns without installing it. Dynamic analysis executes the application in a fully isolated environment, i. e. sandbox, which intervenes and logs low-level interactions with the

system for further analysis. Both the sandbox and the detection algorithms can be deployed in the cloud, providing a fast and distributed detection of suspicious software in a mobile software store akin to Google's Android Market. Additionally, AASandbox might be used to improve the efficiency of classical anti-virus applications available for the Android operating system.

In 2012, *Johnson et. al.* [8] presented a paper in IEEE named *Analysis of Android Applications' Permissions.* In this Paper, mapping of Android application programming interface (API) calls to the required permission(s), if any, for each call. An analysis of Android applications to determine if they have the appropriate set of permissions based on the static analysis of the APK bytecode of each application. This paper indicate that the majority of mobile software developers are not using the correct permission set and that they either over-specify or under-specify their security requirements.

In 2012, Struse *et. al.* [11] presented a paper in proceedings Springer named Permission Watcher: Creating User Awareness of Application Permissions in Mobile Systems. In this paper, they developed Permission Watcher, an Android application which provides users with awareness information about other applications and allow to check on the permission set granted to individual applications.

As they focus on raising user awareness of Android permissions, this leads to the following two limitations:

- Do not cover Permission re-delagation.
- Do not analyze the code nor the behavior of application.

In 2010, Mohammad *et. al.* [13] presented a paper in proceedings ACM named Apex: Extending Android Permission Model and Enforcement with User-defined Runtime Constraints. In this paper, Apex provides framework for Android that allows a user to selectively grant permissions to applications as well as impose constraints on the usage of resources. They also describe an extended package installer that allows the user to set these constraints through an easy-to-use interface

In 2012, Sarma *et. al.* [12] presented a paper in proceedings ACM named Android Permissions: A Perspective Combining Risks and Benefits. In this paper, they investigate the feasibility of using both the permissions an app requests, the category of the app, and what permissions are requested by other apps in the same category to better inform users whether the risks of installing an app is commensurate with its expected benefit.

In 2012, Kern *et. al.* [9] presented a paper in proceedings *UBICOMM* named Permission Tracking in Android. In this paper, they have a closer look at permissions that users grant to apps in Android, a wide-spread operating system for mobile devices like smart phones. They developed tool that allows users to administer permissions of their applications. They enable users to allow or deny permissions at any time.

In 2012, Yarn *et. al.* [10] presented a paper in proceedings ACM named Short paper: enhancing users' comprehension of android permissions. In this paper, they propose to help Android users better understand application permissions through crowdsourcing. In this approach, collections of users of the same application use our

tool to help each other on permission understanding by sharing their permission reviews. They developed a tool called Droidganger which is constructed using two techniques:

- *record/replay*
- *permission suppression.*

Conclusion:

From the given literature this has been listed out that malware increases more rapidly in android smartphones. We have described the various static and dynamic approaches that are used to enhance the security factors and permission mechanisms like Apex, Droidganger, Permission Watcher etc. Hence there is a need for such technique that can reduce the malwares like spyware which monitors the personal information of user and steal their information for their personal use.

References:

[1] Parmjit Kaur, Sumit Sharma, "Google Android A Mobile Platform: A Review. " In *Recent Advances in Engineering and Computational Sciences (RAECS), 2014*, pp. 1-5. IEEE, 2014.

[2] Egele, Manuel, Theodoor Scholte, Engin Kirda, and Christopher Kruegel. "A survey on automated dynamic malware-analysis techniques and tools. " *ACM Computing Surveys (CSUR)* 44, no. 2 (2012): 6.

[3] Vargas, Ruben Jonathan Garcia, Eleazar Aguirre Anaya, Ramon Galeana Huerta, and Alba Felix Moreno Hernandez, "Security controls for Android" In *Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference on*, pp. 212-216, IEEE, 2012.

[4] Vinod, P., R. Jaipur, V. Laxmi, and M. Gaur. "Survey on malware detection methods. " In *Proceedings of the 3rd Hackers' Workshop on Computer and Internet Security (IITKHACK'09)*, pp. 74-79. 2009.

[5] Mohd Afizi, Mohd Shukran, Wan Sharil and Sham Bin Sharif, "Android Augmented Reality System In Malaysia Military Operations – Unit Positions, " in Australian Journal of Basic and Applied Sciences, Vol. 6, Issue 8, p79, Aug2012.

[6] Blasing, Thomas, Leonid Batyuk, A-D. Schmidt, Seyit Ahmet Camtepe, and Sahin Albayrak. "An android application sandbox system for suspicious software detection" In *Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on*, pp. 55-62, IEEE, 2010.

[7] a_pinto "Android Malware 400% increase" [Available online]: http://cybersecurity. mit. edu/2012/11/android-malware-400-increase.

[8] Johnson, Ryan, Zhaohui Wang, Corey Gagnon, and Angelos Stavrou. "Analysis of Android Applications' Permissions. " In *Software Security and Reliability Companion (SERE-C), 2012 IEEE Sixth International Conference on*, pp. 45-46. IEEE, 2012.

[9] Kern, Michael, and Johannes Sametinger. "Permission Tracking in Android. " In*UBICOMM 2012, The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 148-155. 2012.

[10] Yang, Liu, Nader Boushehrinejadmoradi, Pallab Roy, Vinod Ganapathy, and Liviu Iftode. "Short paper: enhancing users' comprehension of android permissions. " In *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, pp. 21-26. ACM, 2012.

[11] Eric Struse, Julian Seifert, Sebastian, Ullenbeck, Enrico Rukzio and Christopher Wolf. “Permission Watcher: Creating User Awareness of Application Permissions in Mobile Systems” ” in proceedings 2012 springer, pp. 65-80. Springer, 2012.

[12] Sarma, Bhaskar Pratim, Ninghui Li, Chris Gates, Rahul Potharaju, Cristina Nita-Rotaru, and Ian Molloy. "Android permissions: a perspective combining risks and benefits. " In *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*, pp. 13-22. ACM, 2012.

[13] Nauman, Mohammad, Sohail Khan, and Xinwen Zhang. "Apex: extending android permission model and enforcement with user-defined runtime constraints. " In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pp. 328-332. ACM, 2010.