Volume 4, Issue 7, July 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Manifest files Analyser System for Detecting Android Malware

Minakshi Ramteke[*]
Computer Science and Engineering
Nagpur Institute of technology
Nagpur, India

Suchit Sapate
Computer Science and Engineering
St. Vincent Pallotti College
Nagpur, India

Prof. Praveen Sen
Computer Science and Engineering
Nagpur Institute of technology
Nagpur, India

Abstract— Android is an open source and fast growing technologies used in tab, mobile devices and Smart phones. The threats of android attacks are increasing with the popularity of smart phone usage. So the many studies going on the mobile security. Once the device infected with the malware, the user suffers from various malicious activities. Malicious activity could be slow response experience of the device; stole the secure information from device, device can be operated remotely and so on. Much research is going on to analyse the malware behaviour so that malware can be detected easily. Now a day's malware behaviour is evolving rapidly so detecting the malware is not an easy task for the researcher.

In this paper we have provided the novel approach for detecting the malware. Our newly proposed approach is very effective and fast because it analyses the manifest file of an Android application. Manifest file necessarily present in every Android Application, which contains essential information about the application. Our proposed method analyses the manifest file to extract the permission required to run the application. This characteristic of the manifest file is a very effective parameter for detecting the malware. This method can be easily combined with other approaches to form a new system which realizes an even more precise detection system.

Keywords— Android, Malware, Manifest.

I. INTRODUCTION

Smart Phone popularity is increasing day by day & Android is an open source platform so it is wildly used is smart phone. But open nature of Android platform makes it prime target of attacker. Attacker misuses the openness of Android platform to spread the malware on the Android device. Numerous researches are still going on for the mobile security. There are many approaches are found but signature based approach is more effective & popular among them. This paper is solely based on signature based approach.

Our newly proposed method for malware detection is more effective than other because this method just analyses the manifest file of Android application. The manifest file must be present for each Android application which contains essential information about the application. Our proposed method analyses the manifest file for extracting the permission required to run that application & permission characteristic of manifest file is very effective parameter for detecting the malware. This approach is very light weighted approach and quick as well.

The main aim of this work is to identify the malicious application on the basis of permissions which are required to execute the application. This work is an attempt to address the following defined problems:

1. Simple and effective way to safely execute and analyse applications
2. Develop a pattern or signature to identify the malicious application.
3. Determine the malwares very efficient and effective way.
4. Application should be very light weighted and fast as well.
5. Signature database can be stored on cloud environment because cloud database should be up-to-date so that newly found malware can easily detect.
6. Application should have capability to remove the malwares.
7. Application should have the provision to kill or stop the running applications.

II. PROPOSED METHOD FOR DETECTING THE MALWARE

Here we proposed a new approach for detecting Android malware by analysing only manifest files. In this approach our application only analyses the manifest file data and extracts the required permission section from it, to categorize the application as benign or malicious. Android applications package is nothing but the zip format package which contains: manifest file, application programs for Dalvik virtual machine (VM), and application resources.

"AndroidManifest.xml" is a manifest file of application, which must be present in all Android applications. It describes the name, version, access rights, referenced library files, and other information of the application. Application programs contains the compiled application program files such as "classes.dex." this format understandable by the Dalvik Virtual Machine. Application resources consist of resources used in the application such as images, icons, and video files [20].

III. COMPONENTS

The method used following components for detecting the malware

1. *Extract Installed Application:*

   This component extracts all the installed application of the device and displays it on the screen. This component also analyses the installed application's manifest file and extract below details from it.

   A. Screens: Screen or layouts installed by the application.
   B. Permission: Permissions created by the application.
   C. Permission Requested: Permissions requested by the application.
   D. Services: Services used by the application to send or receive the background data.
   E. Signature: Authenticate Certificate or signature used by application.
   F. Instrumentation: Name of all instrumentation which monitors all the interaction the system has with application.
   G. Receivers: Receive system wide events by the application.
   H. Features: Features used by the application.
   I. Providers: Data providers used by the application.
   J. Preferences: Configuration preferences of the application.

2. *Manage Filter:*

   This component is used for managing the filters that is signatures on the basis of requested permission of the application. It allows creating or updating the filters by adding/removing the permissions into the local signature database. It also allows viewing the permission in the encrypted format. In this work AES algorithm is used to encrypt or decrypt the signatures.

3. *Cloud database handler:*

   This component is used to upload or download the database on cloud. In this work the google drive is used as a cloud for the demo purpose. Also, it added the google play services library and used google drive API to do all the operations such as database uploading and downloading.

   Files in the Google Drive Android API, represented by the DriveFile interface, are specialized resources with Metadata, a DriveId, and Contents. The binary content of a file is encapsulated in the Contents class which also contains methods for reading from and writing to the file contents.

4. *Task status finder and Killer:*

   This component is used for identifying the running application and it also provides the provision to kill or stop those applications.

5. *Malware Identifier:*

   This is most important component, used for identifying the safe or malicious application on the basis of local database signatures. Basically this component matches the signature permissions with the requested permissions found in the manifest file of the application, if matched, then categorized the application with different colour codding provided while creating the filters or signatures. This component also categorized the application on the basis of total number of activities and services contains in the application. Those categories are

   A. Very Light: If application contains less than 5 activities or services.
   B. Light: If application contains less than 10 but more than 4 activities or services.
   C. Medium: If application contains less than 20 but more than 9 activities or services.
   D. Heavy: If application contains less than 30 but more than 19 activities or services.
   E. Very Heavy: If application contains more than 29 activities or services.

   This component also provides the provision for uninstalling the application.

IV. ANDROID MALWARE DETECTION FLOW

Android malware is detected by the following steps:

[Step 1]: Start the application to scan all the installed application on your device.
[Step 2]: Find all the android application installed on your device.
[Step 3]: Fetch the one application from installed application list and analyse the android applications manifest file.
[Step 4]: Extract the entire permissions list from the manifest file of that application.
[Step 5]: Compare the extracted permission with the database signatures permissions.
[Step 6]: If permission matches then apply the appropriate colour code of that application according to matched signature, which means it's a malware application. Otherwise, don't apply any colour code to the application that means it is a benign application. Then go to step 3 for scanning the entire installed application list.

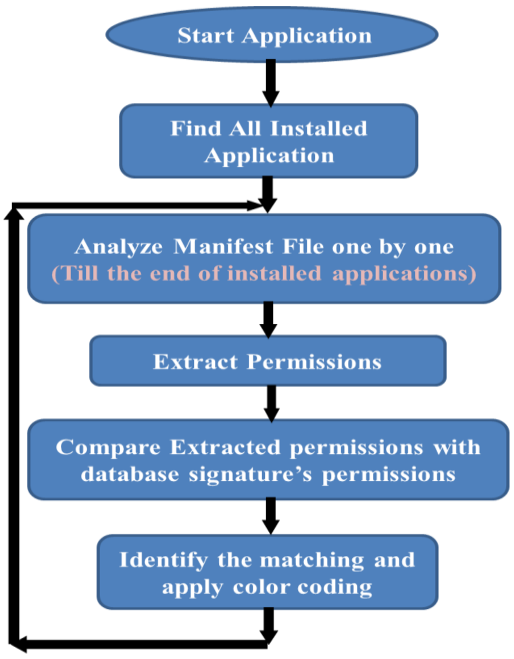
Start Application
Find All Installed Application
Analyze Manifest File one by one
(Till the end of installed applications)
Extract Permissions
Compare Extracted permissions with database signature's permissions
Identify the matching and apply color coding

Figure 1: Android Malware Detection Flow

V. RESULT AND DISCUSSION

This application scans the android device in two phases. In first phase it categorizes the application on the basis of total number of activities and services contains in the application. This is one of the good parameter to identify the heavyweight application which consumes the more resources of the device. Following graph shows the category distribution of sample android device.

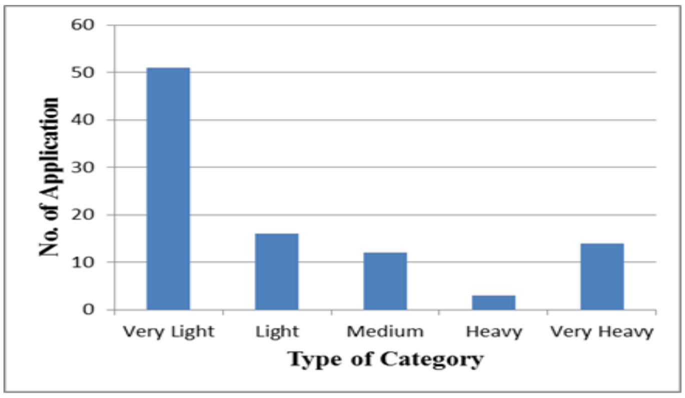

Figure 2: Categories Distribution of Installed Apps

In second phase application scans all the installed application and identifies the malware application on the basis of permissions. This application applies appropriate colour coding to the malware application on the basis of matching signatures. Following table shows some dangerous permission used in the malware applications.

TABLE I: MALWARE DANGEROUS PERMISSION SET

Sr. No.	**Permission**	**Description**
1	WRITE_SETTINGS	Allows an application to read or write the system settings.
2	WRITE_SMS	Allows an application to write SMS messages.
3	WRITE_SECURE_SETTING S	Allows an application to read or write the secure system settings.
4	SEND_SMS	Allows an application to send SMS messages.
5	BROADCAST_SMS	Allows an application to broadcast an SMS receipt notification.
6	PROCESS_OUTGOING_CA LLS	Allows an application to see the number being dialled during an outgoing call with the option to redirect the call to a different number or abort the call altogether.

7	READ_CALL_LOG	Allows an application to read the user's call log.
8	INTERNET	Allows applications to open network sockets.
9	WRITE_EXTERNAL_STORAGE	Allows an application to write to external storage.
10	READ_CONTACTS	Allows an application to read the user's contacts data.
11	READ_SMS	Allows an application to read SMS messages.
12	ACCESS_COARSE_LOCATION	Allows an app to access approximate location derived from network location sources such as cell towers and Wi-Fi.
13	ACCESS_FINE_LOCATION	Allows an app to access precise location from location sources such as GPS, cell towers, and Wi-Fi.

On the basis of above dangerous malware permission this application identifies the malwares application but it also shows some benign application as malware. So user must be smart enough to decide whether identified application is a benign or malware while removing the application.

VI. CONCLUSION

This work proposed a new detection method for Android malware. The main benefit of this method is that it analyses only manifest files to detect malware. This method is applicable to all Android application because Manifest files are required in all Android applications. Likewise, the cost of analysing process is very low because manifest analysis required fewer resources.

The proposed approach analyses only the requested permission of the application to the database so it is very simple and faster than other. The database signatures are created on the basis of android permissions so the structure of database very simple as well. This approach is static approach, and to solve the static approach problem this application adapted the cloud based approach to save the updated signature database. Due to cloud based approach this application can able to detect the newly find malwares on the basis of updated signature database. This method can be easily combined with other approaches to form a new system which realize an even more precise detection system.

REFERENCES

[1]. Patrik Lantz, “An Android Application Sandbox for Dynamic Analysis”, Lunda University, Master’s Thesis at Department of Electrical and Information Technology, Nov 2011.

[2]. Carlos A. Castillo, “An Android Malware Past, Present, and Future", Mobile Security Working Group McAfee, 2011.

[3]. Mamoun Alazab,Sitalakshmi Venkataraman and Paul Watters, An "Towards Understanding Malware Behaviour by the Extraction of API calls", IEEE Second Cybercrime and Trustworthy Computing Workshop, 2010.

[4]. Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steven Hanna, and David Wagner, An "A Survey of Mobile Malware in the Wild", SPSM’11,ACM 978-1-4503-1000-0/11/10, Oct 2011.

[5]. Guillermo Suarez-Tangil, Juan E. Tapiador, Pedro Peris-Lopez, and Arturo Ribagorda, "Evolution, Detection and Analysis of Malware for Smart Devices", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, 2013.

[6]. Étienne Payet and Fausto Spoto, "Static analysis of Android programs", Elsevier, Information and Software Technology, 54 (2012), 2012.

[7]. É. Payet and F. Spoto, "Static analysis of Android program", in: N. Bjørner, V. Sofronie-Stokkermans (Eds.), Proc. of the 23rd International Conference on Automated Deduction (CADE’11), Lecture Notes in Computer Science, vol. 6803, Springer, 2011.

[8]. L. Yan and H. Yin, “Droidscope: Seamlessly reconstructing the os and dalvik semantic views for dynamic android malware analysis,” in Proc. 21st USENIX conf. on Security symp. USENIX Association, 2012.

[9]. Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song and David Wagner, “Android Permissions Demystified”, CCS’11,ACM 978-1-4503-0948-6/11/10, 2011.

[10]. I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, “Crowdroid: behaviorbased malware detection system for android,” in 1st ACM workshop on Security and privacy in smartphones and mobile devices. ACM, 2011.

[11]. MoutazAlazab, VeelashaMoonsamy Lynn Batten, RonghuaTian, and Patrik Lantz “Analysis of Malicious and Benign Android Applications,” IEEE, 32nd International Conference on Distributed Computing Systems Workshops, 2012.

[12]. Dong-Jie Wu, Ching-Hao Mao, Te-En Wei, Hahn-Ming Lee and Kuo-Ping Wu, “DroidMat: Android Malware Detection through Manifest and API Calls Tracing,” IEEE, Seventh Asia Joint Conference on Information Security, 2012.

[13]. G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and J. B. Alis,“Dendroid: A text mining approach to analyzing and classifying code structures in android malware families,” Expert Systems with Applications, July 2013.

[14]. T. Blasing, L. Batyuk, A. Schmidt, S. Camtepe, and S. Albayrak, “An android application sandbox system for suspicious software detection,”in 5th Int. Conf. on Malicious and Unwanted Software (MALWARE 2010). IEEE, 2010.

[15]. W. Enck, P. Gilbert, B. Chun, L. Cox, J. Jung, P. McDaniel, and A. Sheth, "Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones," in Proc. 9th USENIX conf. on Operating systems design and implementation. USENIX Association, 2010.

[16]. Michael Spreitzenbarth,Felix Freiling, Florian Echtler,Thomas Schreck and Johannes Hoffmann, "Mobile-Sandbox: Having a Deeper Look into Android Applications," SAC'13, ACM 978-1-4503-1656-9/13/03, March 2013.

[17]. Nitin Padriya and Nilay Mistry, "Review of Behavior Malware Analysis for Android", IJEIT, Vol. 2, Issue 7, Jan 2013.

[18]. Katsunari Yoshioka, Yoshihiko Hosobunchi, Tasunori Orii and Tsutomu Mastsumoto, "Vulnerability in Public Malware Sandbox Analysis System", IEEE 10th Annual International Symposium on Application and the Internet, 2010.

[19]. Saurabh Chamotra,Rakesh Kumar Sehgal and Raj Kamal, "Honeysand: An Open Source Tools Based Sandbox Environment for Bot Analysis and Botnet Tracking," Special Issue of International Journal of Computer Applications (0975 – 8887) on Communication Security, No.7, March 2012.

[20]. Ryo Sato, Daiki Chiba and Shigeki Goto, "Detecting Android Malware by Analyzing Manifest Files," Proceedings of the Asia-Pacific Advanced Network 2013 v. 36, p. 23-31.

[21]. Nwokedi Idika and Aditya P. Mathur, "A Survey of Malware Detection Techniques", Department of Computer Science, Purdue University, West Lafayette, IN 47907, February 2, 2007.

[22]. Naresh Kumar and Muhammad Ehtsham Ul Haq, "Penetration Testing of Android-based Smartphones," Master of Science Thesis in the Programme Networks and Distributed Systems, Chalmers University of Technology, University of Gothenburg, Department of Computer Science and Engineering, June 2011.

[23]. "Manifest.permission", Android 4.4 r1, 07 Jul 2014, http://developer.android.com/reference/android/package-summary.html.

[24]. Wikipedia, "Android (operating system)," Wikimedia Foundation, Inc., 7 July 2014, http://en.wikipedia.org/wiki/Android_operating_system.