Survey on Android Application's Permissions and User's Data Protection

Shrishchandra Wakchaure[1], Sumiran Hadap[2], Ashish Thombre[3], Sandip Marepalle[4], Nilam Patil[5]
[1, 2, 3, 4, 5] Department of Computer Engineering
[1, 2, 3, 4] Students,DYPCOE, Pune University, Pune, Maharashtra, India
[5]Assistant Professor,DYPCOE, Pune University, Pune, Maharashtra, India

Abstract-*Most existing mobile malware detection methods (e.g., Kirin and Droid-Mat) are designed based on the resources required by malwares (e.g., permissions, application programming interface (API) calls, and system calls). Existing malware detection methods are able to identify inter-app vulnerabilities. Nowadays, android's permission system is used to inform users about the risks of installing applications. World Wide users have begun downloading an increasingly large number of mobile phone applications in response to advancements in Android technologies and wireless networks. Increased in number of applications results in a greater chance of installing malwares. When a user installs an application, he or she has the opportunity to review the android applications permission requests and cancel the installation if the permissions are objectionable or excessive. COVERT, a compositional analysis of Android inter-app vulnerabilities. COVERT classify application into benign or malicious. This method capture the interactions between mobile apps and Android system. In our research, we find out that there are many inter-app vulnerabilities such as collusion attack and privilege escalation chaining which are used to increase privilege of app. Many of this inter-app vulnerabilities are yet to be discover.*

Keywords-Android malware, Android security, collusion attacks, feature-based, Inter-App vulnerabilities, Privilege Escalation Attacks, Smartphone security.

I. INTRODUCTION

Today smartphones are ubiquitous; smartphones have become the inseparable part of people's life by providing various services and different functionalities. The ubiquity of smartphones and our growing dependency on mobile apps are making us more vulnerable to security attacks than ever before. Telecommunication as a technology is constantly evolving. Recently it has adopted Fourth generation (4G) wireless communication and handsets with advanced microprocessors. Also, number of applications for this android platform are increasing with the increasing number of technologies used in adapting smartphones and Android mobile handsets. User may install various applications without checking their security vulnerabilities. It may be installed either from trusted or untrusted sites. It may cause harm to user's privacy. There are many ways of security violation in Android system by which security can be compromised. In our survey, we have discovered many vulnerabilities which give rise to security violation in android applications due to permission leakage. Mostly these vulnerabilities are inter-app vulnerabilities which are not fully detected by existing tools. In recent years, many tools are launched which are able to discover or handle the inter-app vulnerabilities. Our research shows that there are many tools to detect and eliminate the user's data and privacy vulnerabilities in android system but these tools fails to detect the inter-app vulnerabilities which are raised due to inter-communication or interaction between two or more applications such as collusion attacks and privilege escalation chaining.

II. SURVEY ON APPLICATION PROTECTION MECHANISM

A. Droid-Mat:-

Recently, Android malware is spreading rapidly causing threats to android users, especially those repackaged Android malware. Understanding Android malware using dynamic analysis can provide a comprehensive view, it is still subjected to high cost in deployment and manual efforts in investigation. In this, static feature-based mechanism to provide a static analyst paradigm for detecting the Android malware had been proposed. This mechanism considers the static information including permissions, deployment of components, Intent messages passing and API calls for characterizing the Android applications behavior. In order to recognize different intentions of Android malware, different kinds of clustering algorithms can be applied to enhance the malware modelling capability. Besides, a mechanism was proposed and a system was developed, called *Droid Mat*. Firstly, the *Droid Mat* extracts the information (e.g., requested permissions, Intent messages passing, etc.) from each application's manifest file, and regards components (Activity, Service, Receiver) as entry points drilling down for tracing *API Calls* related to permissions. Then, it applies K-means algorithm which enhances the malware modelling capability. *Singular Value Decomposition (SVD)* method decides the

number of clusters on the low rank approximation. Finally, it useskNN algorithm and classifies the application as benign or malicious. The experiment result shows that the recall rate of our approach is better than others, which focuses on Android malware analysis. Also, *Droid-Mat* is efficient since it takes only half of time than Andro-guard to predict 1738 applications as benign or malicious [2]

This extracts the full information about the application including its manifest file and disassembly codes for further analysis. The fig below explains Droid-Mat static functional behavior analysis for android malware detection.[2]

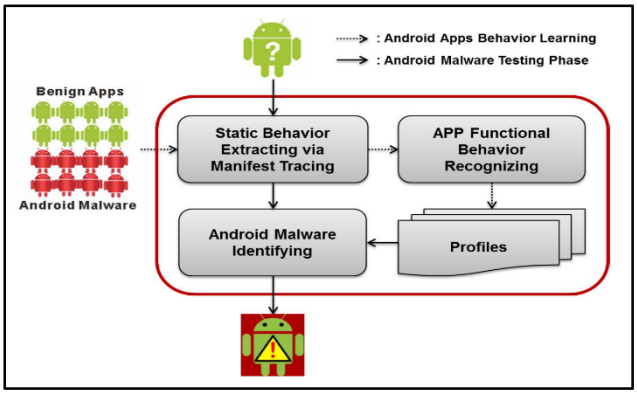
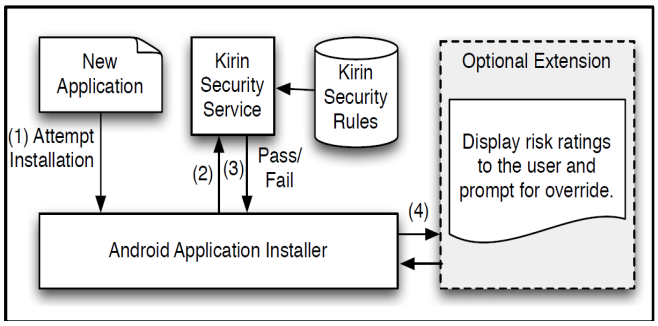


Fig.1 Droid-Mat Architecture [2]

B. The Kirin security

Users are downloading an increasingly large number of mobile phone applications in response to advancements in handsets and wireless networks. This increased number of applications results in a greater chance of installing Trojans like malwares. In this paper, they proposed the Kirin security service for Android, which performs lightweight certification of applications to mitigate malware at the time of installation. Kirin certification uses security rules, which are nothing but the templates designed to conservatively match undesirable properties in security configuration bundled with applications. [1]



ig. 3 Kirin based software installer [1]

They use a variant of security requirements engineering techniques to perform an in-depth security analysis of Android and produce a set of rules that match malware characteristics. In most popular applications downloaded from the official Android Market, Kirin found 5 applications that implement dangerous functionality and hence should be installed with extreme caution. Upon close inspection, another five applications asserted dangerous rights, but were within the scope of reasonable functional needs. These results indicate that security configuration bundled with Android applications provides practical means of detecting malware. [1]

Security violation: - For several years, researchers have been studying mobile phone security. At first, mobile malware was just a concept. Over time, mobile malware has become a real threat in our daily life. We survey the state of modern mobile malware in the wild to illuminate the current threat model and suggest future directions. Malicious apps are using clever ways to bypass existing security mechanisms which are provided by Android OS as well as anti-malware products such as code obfuscation, stealth techniques, dynamic execution, repackaging and encryption.

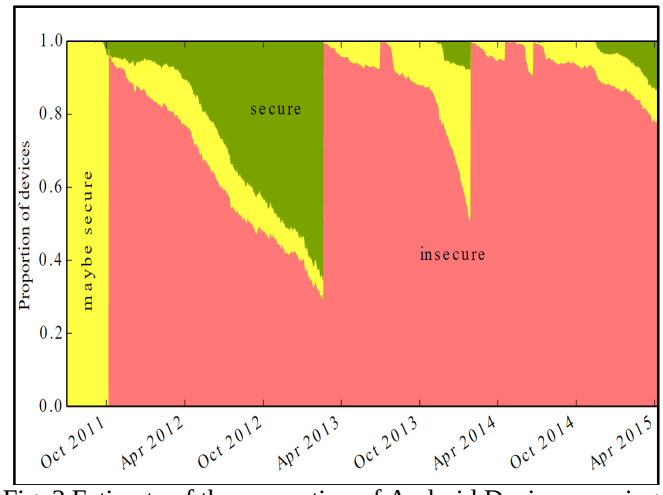


Fig. 2 Estimate of the proportion of Android Devices running insecure over time [8]

III. PRIVACY SECURITY THREATS

The number of malicious Apps of Android is increasing year by year due to the Open Source of the Android operating system and the disordered Android App market. Various Threats divided the sensitive data into four categories: Device Resources, User data, System Information and Application Data. Each of these categorical threats are analyzed as follows.

A. Device Resources

Lots of malicious sensor Apps spring up in recent years. Mobile devices equip with GPS, NFC, camera equipment and other sensors which enable Apps to accomplish complex functions and services, such as mobile phone navigation. However, it was found that hardware imperfections make each accelerometer sensor chip respond differently to the same motion stimulus. NFC is widely used in many fields such as access control, payment, and ticketing.

B. User data

While using the basic service of mobile devices, including the short message, contact list, phone records, etc., user data is produced. As per Felt et al., the most common malicious behavior is the stealing of personal information of users. Additionally, most instant messaging software request access to the contact list and use the address book to recommend the other contacts to users. Although it becomes convenient for attackers as they can utilize this method to automatically obtain the user's privacy information.

C. System Information

System Information consist of various identity information, status information, such as phone number, IMEI (GSM), MEID, ESN (CDMA), Android ID, and MAC Wi-Fi address etc. Some information can uniquely identify a mobile device, namely, identify a user. It was found that ACCESS_WIFI_STATE permission can not only identify users with MAC address, but also obtain their coarse-grained location information without requesting the location permission.

D. Application Data

Applications create numerous data and cookies which directly relate to the user's interest. Although Android applies the sandbox mechanism to ensure the isolation between the Apps, there are still some means for malware to gather and analyses these data. Nowadays, Web-View is by most Apps to display the HTML content within them. One feature of Android is that it provides a way for JavaScript in a Web-View invoking Android App code, if Web-View is enabled by the App. This allows the web page to access functionality and data exposed by the App, which undoubtedly increases Apps' attack surface.

IV. WAYS TO LEAK PRIVACY INFORMATION

A. Permission escalation

Malicious Apps can leverage the vulnerabilities in Android system or the privileged abilities exposed by Apps. This is done to escalate their permission, and then to operate users' sensitive information. Xing et al. brought to light a new type of security-critical vulnerabilities, called "Pileup" flaws, through which a malicious app can strategically declare a set of privileges and attributes on a low-version operating system. And then it waits to escalate its privileges when the OS is upgraded.

B. Collusion attack

Attackers accomplish malicious behavior by colluding Apps, they indirectly escalate their permissions. Collusion attack can escape those detection technologies which are designed for a single App. It was implemented and analyzed a number of covert and covert communication channels that enable Apps to collude, which revealed the seriousness of collusion attack.

TABLE I [10]

Android Permission Group	**Permission Detail**
android.permission - group.PHONE	android.permission.CALL_ PHONE android.permission.READ_ CALL_ LOG android.permission.WRITE_ CALL_LOG android.permission.PROCESS_ OUTGOING_CALLS android.permission.READ_ PHONE_STATE
android.permission -group. CONTACTS	android.permission.READ_ CONTACTS android.permission.WRITE_ CONTACTS
android.permission -group. MICROPHONE	android.permission.RECORD_ AUDIO

android.permission -group.SMS	android.permission.SEND_SMS android.permission.RECEIVE_ SMS android.permission.READ_ SMS android.permission.RECEIVE_ MMS android.permission.SEND_ MMS android.permission.RECEIVE_ WAP_PUSH
android.permission -group. CALENDER	android.permission.READ_ CALENDER android.permission.WRITE_ CALENDER
android.permission -group.CAMERA	android.permission.CAMERA
android.permission -group.SENSORS	android.permission.BODY_ SENSORS
android.permission -group. LOCATION	android.permission.ACCESS_ FINE_LOCATION android.permission_ACCESS_ COARSE_LOCATION
android.permission -group.STORAGE	android.permission.READ_ EXTERNAL_STORAGE android.permission.WRITE_ EXTERNAL_STORAGE

V. PROPOSED WORK

In the proposed method for future work, APK extraction tool is used to extract permission of android apps. Using FAST algorithm, the irrelevant features are removed and there after it create MST from relative ones. Lastly it partitions the MST and select the representative feature which will get labelled after. For classification SVM is used on extracted permission of android application. On that basis training and testing will be applied on these permissions. Implementing SVM for classification. Perceptual constraints, functional constraints are applied on Feature extraction. It is also important that how well the system remove malicious applications.

VI. CONCLUSION

Data is analyzed and using that data the system is proposed to identify the malicious apps. It was found out that there are many tools, mechanisms exist to analyze or detect the vulnerabilities but were unable to find inter-app vulnerabilities occurring due to inter-app communication between two apps. Some tools are used to detect inter-app vulnerabilities like COVERT and their result shows significant amount of inter-app vulnerabilities related to permission leakage. Our proposed system mainly handle these vulnerabilities.

REFERENCES

[1] W. Enck, M. Ongtang, and P. McDaniel, “On lightweight mobile phone application certification,” in Proc. ACM Conf. Comput. Commun. Security, 2009, pp. 235–245.USA: Abbrev. of Publisher, year, ch. x, sec. x, pp. xxx–xxx.

[2] Dong-Jie Wu, Ching-Hao Mao, Te-En Wei, Hahn-Ming Lee, Kuo-Ping Wu, “Droid-Mat: Android Malware Detection through Manifest and API Calls Tracing” in 7th Asia JCIS,2012

[3] Android Manifest, http://developer.android.com/guide/topics/manifest/manifestt-intro.html

[4] Android Security,http://developer.android.com/training/articles/security- tips.html

[5] AndroidPermission,https://android.googlesource.com/platform/frameworks/base/+/master/core/res/AndroidManifest.xml

[6] Android API Guide - Permission, http://developer.android.com guide/topics/manifest/permissionelement.html

[7] Malicious apps hosted on Google store turn androidphone into zombies http://arstechnica.com/gadgets/2012/05/malicious-appshostedin-google-market-turn-android-phones-into-zombies/

[8] http://androidvulnerabilities.org/graph

[9] Google now scanning for android apps for malwarehttp://www.cnet.com/news/google-now-scanning-android-apps-formalware/

[10] Dr. K. V. Kulhalli, Nitesh Arun Patil, A Survey: Inter-App Permission Leakage on Android Devices, Volume 5, Issue 9, September 2015 ,International Journal of Advanced Research in Computer Science and Software Engineering

[11] http://arstechnica.com/gadgets/2012/05/malicious-appshostedin-google-market-turn-android-phones-into-zombies/

[12] Iker Burguera and Urko Zurutuza, Simin Nadjm-Tehrani, "Crowdroid: Behavior-Based Malware Detection System for Android"

[13] Asaf Shabtai,, Uri Kanonov, Yuval Elovici, Chanan Glezer, Yael Weiss, "Andromaly: a behavioral malware detection framework for android devices", January 2011, Springer Science+Business Media

[14] William Enck, Peter Gilbert, Byung-Gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, Anmol N. Sheth, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones"

[15] David Barrera , H. Güne¸s Kayacık , P.C. van Oorschot , Anil Somayaji , "A Methodology for Empirical Analysis of Permission-Based Security Model and its Application to Android"

[16] Borja Sanz, Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero and Pablo Garcia Bringas, "On the Automatic Categorisation of Android Applications"

[17] Borja Sanz, Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero, Pablo Garcia Bringas, and Gonzalo _Alvarez, "PUMA: Permission Usage to detect Malware in Android"

[18] Kevin Benton, L. Jean Camp, Vaibhav Garg, "Studying the Effectiveness of Android Application Permissions Requests", Fifth International Workshop on Security and Social Networking 2013, San Diego (18 March 2013)

[19] Rohit Kale, Prof. P D. Lambhate, "Malware security for Android Components using Layer permission", Volume 3, Issue 5, May 2015, International Journal on Recent and Innovation Trends in Computing and Communication

[20] Adrienne Porter Felt, Helen J. Wang, Alexander Moshchuk, "Permission Re-Delegation: Attacks and Defences"

[21] Mrs. Gunjan Kapse, Prof. Aruna Gupta, "Detection of Malware on Android based on Application Features", Vol. 6 (4) , 2015 International Journal of Computer Science and Information Technologies

[22] Balaji Baskaran and Anca Ralescu, "A Study of Android Malware Detection Techniques and Machine Learning" 2016, Modern Artificial Intelligence and Cognitive Science Conference

[23] Karim O. Elish, Danfeng (Daphne) Yao, and Barbara G. Ryder, "On the Need of Precise Inter-App ICC Classification for Detecting Android Malware Collusions"

[24] Hamid Bagheri, Alireza Sadeghi, Joshua Garcia and Sam Malek "COVERT: Compositional Analysis of Android Inter-App Permission Leakage", 2015, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING