International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 6 Issue: 3 199 - 204

Review of Malware Detection in Android Applications using Dynamic Analysis

Rakhi Wadhai
Department of Computer Science & Engineering
Swami Vivekanand College of Engineering, Indore

Mr. Archit Shukla
Department of Computer Science & Engineering
Swami Vivekanand College of Engineering, Indore

Abstract— Today Android has the biggest market share as compared to other operating system for smart phone. As users are continuously increasing day by day the Security is one of the main concerns for Smartphone users. As the features and power of Smartphone are increase, so that they has their vulnerability for attacks by Malwares. But the android is the operating system which is more secure than any other operating systems available for Smart phones. The Android operating system has very few restrictions for developers and it will increase the security risk for end users. In this paper we have reviewed android security model, application level security in android and its security issues.

Keywords- *Android, vulnerability, malwares, smart phones*

*****

I. INTRODUCTION

The smart phone devices are uses in a range of individual to large enterprises. It will be use for both personal and professional purpose smart phones have become the new personal computer. Consistent performance and ease of handling of the device lets you perform most of the operations often done on a Pc's. These mobile devices are being used not only for just making calls or messaging, but also for interacting with social networking Websites and sometimes performing sensitive financial transactions. There are many operating systems available for the smart phones, one of this is The Android operating system.

Android is a modern mobile platform which is designed to be truly open source. The Android applications can uses advanced level of both hardware and software as well as local and server data, through this platform developer bring innovation and value to consumers. Android platform must have security mechanism to ensure security of user data, information, application and network [1].

To provide security in Open source platform needs strong and rigorous security architecture. The Android is designed with multi-layered security which will provides flexibleness needed by an open source, whereas providing protection for all users of the platform designed to a software stack android includes a middleware and core application as a complete [2].

Android architecture is designed with keep ease of development ability for developers. The Security controls have designed to minimize the load on developers. The Developers have to simply work on versatile security controls because developers are not familiar with securities that apply by defaults on applications.

A. Android Platform Security Architecture: Android seek to be the most secure and usable operating system for mobiles by repur-posing classical operating system security controls to protect user data as well as system resources and provides isolation in application.

Android provides following security features to achieve these objectives are first robustness at the operating system level through the Linux kernel. second compulsory application sandbox for all applications. third secure inter process communication. Fourth application signing and fifth application defined permission and user have to grant permissions.

Figure-1 Android Architecture

Figure 1 summarizes security components and considerations at the various levels of the Android Architecture. Each component assumes that component below is properly secured. Except some Android operating system code running as root and all process run above the Linux Kernel is restricted by the Application Sandbox.

B. Security in Android

1. Android is open source platform, developers will work along to enhance it[1].
2. Android platform is multitasking software; therefore no application will gain critical access to the components of OS[3].
3. Android platform is UNIX based operating system that is the most secure operating system[1].
4. The developers need a unique signature to publish their application on market[4].
5. Users will report a possible security flaw through their Google account.
6. All applications on android need permission from the user at the time of installation.

*C. **Security Issues faced by Android***

Android is not secure as it appear, even when such robust security measures. There are several security problems faced by the android, which are given below:

1. Android has no security scan over the apps being uploaded on its market.
2. There are some apps which can exploit the services of another app without permission request.
3. Android's permission security model provides power to user to make a decision whether an app should be trusted or not.
4. The Open Source is available to legitimate developers as well as hackers too. So that the Android framework cannot be trusted when it comes to develop critical systems.
5. The Android operating system developers clearly state that they are not responsible for the security of external storage.
6. Any app on the android platform will access device data just like the GSM and SIM marketer Ids while not the permission of the user.

II. LITERATURE SURVEY

W. Enck, D. Octeau, P. McDaniel and S. Chaudhuri presented review paper on "a study of Android application security". Introduces the ded decompiler which generate android application source code directly from its installation image. Also they design and execute a horizontal study of smart phone applications based on static analysis of 21 million lines of recovered code. This analysis uncovered pervasive use, misuse of personal or phone identifiers and deep penetration of advertizing and analytics networks [5].

S. Powar, Dr. B. B. Meshram, review on "Android security framework", Describes android security framework. Raising exposure of open source Smartphone is increasing the security risk. The android provide a basic set of permissions to secure smart phone. The method to make Android security mechanism more versatile and the current security mechanism is too rigid. The user has only two options at the time of application installation first allow all requested permissions and second deny requested permissions leads to stop installation [6].

S. Kaur and M. Kaur present "implementing security on Android application". They have described how security can be improved in android system for users to safely use the android smart phones [2].

S. Smalley and R. Craig presented in "Security Enhanced (SE) Android: Bringing Flexible MAC to Android". The android software stack for mobile devices enforces and defines its own security model for apps through its application-layer permissions model. but, at its foundation, the android depends upon the UNIX operating system kernel to shield the system from malicious or imperfect apps and to isolate apps from each other. Now, android leverages UNIX operating system discretionary access control (DAC) to enforce these guarantees. although the notable shortcomings of DAC. this paper, motivates and describe their work to bring flexible mandatory access control (MAC) to Android by enabling the effective use of Security Enhanced Linux (SELinux) for kernel-level MAC and by developing a set of middleware MAC extensions to the Android permissions model [7].

P. Gilbert, W. Enck, A.N. Sheth , L.P. Cox, B.G. Chun, J. Jung and P. McDaniel presented "TaintDroid: An Information-Flow Tracking System for Real-time Privacy Monitoring on smart phones". Currently smart phone operating systems often fail to provide users with sufficient control over and visibility into how third-party applications use their private stuff. They address these shortcomings with TaintDroid, the system-wide dynamic taint tracking and analysis system capable of at the same time tracking multiple sources of private data. The TaintDroid display real-time analysis by leveraging Android's virtualized execution environment and Monitoring private data to inform use of third-party applications for phone users and valuable input for Smart phone security service firms seeking to identify misbehaving applications [8].

B. J. Berger, M. Bunke and K. Sohr presented an android security case study with Bauhaus. In this paper analysis, they discovered that firms and corporation now uses security software for code analysis to discover security problems in application. They carried out a case analysis on android based mobile in cooperation with a security expert and employed the reverse engineering tool-suite Bauhaus for security assessment. During the investigation they found some inconsistencies in the implementation of the Android security concept. Based on the case study, they suggest several research topics in the area of reverse engineering that would support a security analyst during security assessments [9].

M. Ongtang, S. McLaughlin, W. Enck and P. McDaniel review on "Semantically Rich Application-Centric Security in Android". In this, they have augment the existing android operating system with a framework to meet security requirements and they proposed secure application interaction, improved infrastructure that governs install-time permission assignment and their run-time use as dictated by application provider strategy. Secure application interaction provides necessary utility for applications to assert and control the security decisions on the android platform[10].

H. G. Schmidt, A.D. Schmidt, J. Clausen, A. Camtepe, S. Albayrak, K. Ali Yüksel and O. Kiraz review on "enhancing security of Linux-based android devices". presents an analysis of security mechanism in Android Smart phones with a focus on Linux. Results of their discussion can be applicable for android as well as Linux-based smart phones. They analyzed android structure and the Linux-kernel to check security functions. They also review well accepted security mechanisms and tools which could increase device security. And they provided details on how to adopt these security tools on Android platform and overhead analysis of techniques in terms of resource usage [11].

C. Marforio, A. Francillon, S. Capkun study on 'application collusion attack on the permission-based security model and its implications for modern smartphone systems'. In this they show technique in which permission based mechanisms are used on mobile platforms allows attacks by colluding applications that communicate over explicit and covert communication channel. These bugs for security allow applications to indirectly execute operations that those applications based on their declared permissions that should not be able to carry out. Example operations include disclosure of user's private data (e.g., phone book and calendar entries) to remote parties by applications that do not have direct access to such data or cannot directly establish remote connections. They more showed that on mobile platforms users are not aware of possible implications of application collusion quite the contrary users are implicitly lead to believe that by approving the installation of each application independently based on its declared permissions and that will limit the damage that an application can cause [12]. They show that this is not correct and that application permissions should be displayed to the users differently reflecting their actual implications.

A. Warg ,M. Lange, S. Liebergeld, A. Lackorzynski, M. Peter represented 'L4Android: a generic operating system framework for secure smart phones'. They present a generic operating system framework that overcomes the need of hardware extensions to provide security in smart phones. They bind smart phone operating system in a virtual machine; this framework allows highly secure applications to run side-by- side with the VM. Which is based on a state-of-the-art micro-kernel that ensures isolation between the virtual machine and secure applications [13]?

T. Luo, H. Hao, W. Du, Y. Wang, and H. Yin work on "attacks on WebView in the android system". Web-View is an important element in android platforms, enabling smart phones and tablet apps to insert a simple but powerful browser. To achieve a much better interaction between apps and their embedded browsers, WebView provides range of APIs, permitting code in apps to invoke the JavaScript code within pages which intercept their events and modify those events. Using these features apps will become customized browsers for their required web applications. Now, within the android market, 86 % of the top twenty most downloaded apps in ten various classes use WebView14. The architecture of WebView changes the landscape of the web particularly from the security viewpoint. Two essential component of the Web's security infrastructure are weakened if Web-View and its APIs are used: the Trusted Computing Base (TCB) at the client aspect and therefore the sandbox protection enforced by browsers. Resulting several attacks may be launched either against apps or by them [14].

D. Barrera, H. Güne¸ S. Kayacık, P.C. van Oorschot, A. Somayaji discuss on 'a methodology for empirical analysis of permission-based security models and its application to android'. According to paper, the proposed methodology is of independent interest for visualization of permission based systems beyond current Android-specific empirical analysis. Authors provide some discussion identifying potential points of improvement for the android permission model, trying to augment quality where required without increasing number variety of permissions or overall difficulty [15].

C. Gibler, J. Crussell, J. Erickson and H. chen case learn on 'AndroidLeaks: automatically detecting potential privacy leaks in android applications on a large scale'. Under this published, they have presented a static analysis framework for automatically finding potential leaks of sensitive data in android applications on a large scale.

AndroidLeaks severely reduces the number of applications and the number of traces that a security auditor must verify manually [16]

I. Burguera, U. Zurutuza, S. Nadjm study on 'Crowdroid: behaviour-based malware detection system for Android'. In this paper, they used earlier approaches for dynamic analysis of application behaviour for detecting malware in the android. The detector is embedded in framework for collection of traces from limitless number of real users. This framework has been demonstrated by analysing information composed in the central server using two sorts of data sets: those from artificial malware created for test functions, and people from real malware found in the globe. The technique is shown to be an effective means of analytic the malware and alerting the users of a downloaded malware. This method is avoiding the spreading of a detected malware to a larger scales [17].

M.L. Polla, F. Martinelli and D. Sgandurra discuss "a survey on security for mobile devices". This surveys the vulnerabilities and security solutions over year 2004-2011, this targets high-level attacks on user applications. In this they cluster existing approaches aimed to securing mobile devices against these kinds of attacks into many categories on the basis of the detection architectures, principles, collected information and operating systems. They mainly focus on IDS-based models and tools. This categorization aim to provide a simple and concise view of the underlying model adopted by each approach [18].

W. B. Tesfay, T. Booth, and K. Andersson review 'reputation based security model for android applications'. They planned a cloud based reputation security model as a solution which greatly mitigates the malicious attacks targets the Android market[19] . This way of security solution uses unique user id (UID) which is assign to each application in the android platform. This type of model stores the status of Android applications in an anti-malware providers cloud (AM Cloud). In the investigational results witness that the proposed model can identify the reputation index of a given application and its potential of being risky or not[20].

A. D. Schmidt, S. A. Camtepe, T. Blasing, L. Batyuk and S. Albayrak analysed 'an android application sandbox system for suspicious software detection'. They project an AASandbox (Android Application Sandbox) which performs static as well as dynamic analysis on android applications to detect suspicious applications. The Static analysis scans the software packages for malicious patterns without installing it along with the dynamic analysis executes the app in an isolated environment, i.e. sandbox, that intervenes and logs low-level interactions with the system. Either the sandbox or the detection algorithms can be deployed in the cloud, providing a quick and distributed detection of suspicious application in an app store similar to Google's Play Store .The AASandbox might be used to improve the anti-virus apps available for the android devices [21].

T. Vidas, D. Votipka, N. Christin discuss on 'all your droid are belong to us: a survey of current android attacks'. According to this they look to Android as a specific instance of mobile computing. They all first discuss the Android security model and then some potential weaknesses of the model. Then they should provide taxonomy of attacks to the platform established by real attacks that in the end guarantee privileged access to the device [22].

S. Holla, M. M Katti study on 'Android based mobile application development and its security'. In this they discuss a layered approach for android application development where they can build up application which downloads data from the server and also an Android Application Sandbox (AASandbox) that is able to perform both the static and the dynamic analysis on Android programs who automatically detect suspicious applications [23].

A. Pretschner , D. Feth studied on'flexible data-driven security for android'. They should propose an improved security system beyond the standard permission system. This is possible to implement complex policies that are built on temporal, cardinality, and spatial conditions in the related system. The Enforcement can be done by means of modification or inhibition of certain events. Leveraging recent advances in information flow of tracking technology, policies can also pertain to data rather than single representations of that data [24].

G. Portokalidis, P. Homburg, K. Anagnostakis, and H. bos represented 'Paranoid Android: versatile protection for smartphones'. They suggest a solution in which security checks are applied on remote security servers that host exact replicas of the phones in virtual environments. The servers are not subject to equivalent constraints, permitting user to use multiple detection techniques at the same time. They developed a prototype of this security model for android phones, and show that it is each practical and scalable: they generate no more than 2KiB/s and 64B/s of trace data for high-loads and idle operation respectively, and are able to support quite 100 replicas running on one server[25].

A.D. Schmidt and S. Albayrak represented paper on 'malicious software for smartphones'. They discuss a list of the most common behavior patterns and investigate possibilities how to exploit the given standard Symbian OS API for additional malware functionalities [26].

J. Cheng, S. H.Y. Wong, H. Yang and S. metal present 'SmartSiren: virus detection and alert for smart

phones'. They represented SmartSiren collects the communication activity information from the smart phones, and performs joint analysis to discover both single-device and system-wide abnormal behaviors. They were used a proxy-based design to load process load from resource constrained smart phones and simplify the collaboration among smart phones. Once a potential virus is detected then the proxy quarantines the natural event causation targeted alerts to those directly vulnerable smart phones. After that they have demonstrated feasibleness of SmartSiren through implementation on Dopod 577w smart phone, and evaluated its effectiveness victimization simulations driven by 3-week SMS traces from a national cellular carrier [27].

A.D. Schmidt, R. Bye, H.G. Schmidt, J. Clausen, O. Kiraz, K. Yuksel, A. Camtepe, and S. Albayrak, review on 'static analysis of executables for collaborative malware detection on android'. Since Smart phones become popular for sensitive information and apps, improved malware detection mechanisms are necessary complying with the resource constraints28. The contribution of this review is two fold . Firstly, they execute static analysis on the executables to extract their operate calls in android environment using the command readelf. Method call lists are matched with malware executables for classifying them with part, Nearest Neighbor Algorithms and Prism .And the Second, they present a cooperative malware detection approach to improve result [28].

G. Dini, F. Martinelli, A. Saracino, and D. Sgandurra, discuss on 'MADAM: a multi-level anomaly detector for android malware'. In this analysis, MADAM can monitors android at the kernel-level and user-level to notice real malware infections using machine learning techniques to differentiate between normal behaviors and malicious ones. The primary prototype of MADAM is able to notice several real malware found in the world. The device is not affected by MADAM due to the low range of false positives generated after the training phase[29].

A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, Yael Weiss, analysis 'Andromaly: a behavioral malware detection framework for android devices'. The proposed framework realizes a Host- based Malware Detection System that continuously monitors various features and events obtained from the mobile device and apply Machine Learning anomaly detectors to classify the collected data as normal or abnormal. They developed four malicious applications and check Andromaly's ability to detect new malware based on samples of known malware. They evaluated many combinations of anomaly detection algorithms, feature choice methodologies in order to find out the combination that yields the best performance in detecting new malware on android[30].

III. PROPOSED WORK

Proposed System Working:

1. The user will try to install an application to his/her device
2. The application would be sent to our cloud server
3. The cloud server would check the application by decompiling it
4. Once decompile the application would be check for Malware
5. If Malwares are found the application will not be installed
6. If no Malware found then application will be installed

Diagrammatic Representation

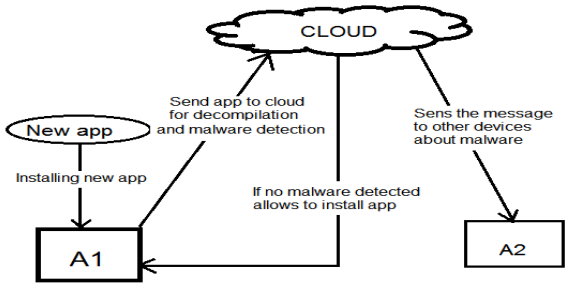

Fig.2. Diagrammatic view of proposed System

IV. CONCLUSION

Now days more than 1 million Android device are activated. The android has very few restrictions for developer which will increases the security risk for end users. In this paper we have reviewed security issues in the Android based Smart phones. The integration of technologies into an application certification process requires overcoming logistical and technical challenges. The android provides more security than other mobile phone platforms.

REFERENCES

[1] Android Open Source Project. Android Security Overview. http://source.android.com/devices/tech/security/index.html(2013)

[2] Kaur S. and Kaur M., Review Paper on Implementing Security on Android Application, Journal of Environmental Sciences, Computer Science and Engineering & Technology, 2(3), (2013)

[3] Android Open Source Project Security and permissions http://developer.android.com/guide/topics/security/ permissions .html. (2013)

[4] Android Open Source Project Publishing on GooglePlay http://developer.android.com/distribute/google play/ publish /preparing.html. (2013)

[5] Enck W., Octeau D., McDaniel P. and Chaudhuri S., A Study of Android Application Security, The 20th USENIX conference on Security, 21-21, (2011)

[6] Powar S., Meshram B. B., Survey on Android Security Framework, International Journal of Engineering Research and Applications, 3(2), (2013)

[7] Smalley S. and Craig R., Security Enhanced (SE) Android BringingFlexibleMACtoAndroid, www. Internetsociety.org/sites/default/files/02_4.pdf. (2012)

[8] Enck W., Gilbert P., Chun B.G., Cox L.P., Jung J., McDaniel P. and Sheth A.N., TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones, 9th USENIX Symposium on Operating Systems Design and Implementation. (2010)

[9] Berger B.J., Bunke M., and Sohr K., An Android Security Case Study with Bauhaus, Working Conference on Reverse Engineering, 179–183 (2011)

[10] Ongtang M., McLaughlin S., Enck W. and McDaniel P., Semantically Rich Application-Centric Security in Android Computer Security Applications Conference, 340–349 (2009)

[11] Schmidt A.D., Schmidt H.G., Clausen J., Camtepe A., Albayrak S. and Yuksel K. Ali and Kiraz O., Enhancing Security of Linux-based Android Devices,http://www.dai-labor.de/fileadmin/files/publications/lk2008-android _security. pdf (2008)

[12] Marforio C., Francillon A. and Capkun S., Application Collusion Attack on the Permission-Based Security Mode and its Implicatios for Modern Smartphone Systems,ftp://ftp.inf.ethz.ch/doc/tech-reports/7xx/724.pdf (2013)

[13] Lackorzynski A., Lange M., Warg A., Liebergeld S., Peter M., L4Android: A Generic Operating System Framework for Secure Smartphones, 18th ACM Conference on Computer and Communications Security, 39-50 (2011)

[14] Luo T., Hao H., Du W., Wang Y. and Yin H., Attacks on WebView in the Android System, 27th Annual Compute Security Applications Conference, 343-352 (2011)

[15] Barrera D., Güne H., Kayacık S., Oorschot P.C. van and Somayaji A., A Methodology for Empirical Analysis of Permission-Based Security Models and its Application to Android, 17th ACM conference on Computer and communications security, 73–84 (2010)

[16] Gibler C., Crussell J., Erickson J. and Chen H., Android Leaks: Automatically Detecting Potential Privacy Leaks In Android Applications on a Large Scale, 5th international conference on Trust and Trustworthy Computing, 291-307 (2012)

[17] Burguera I., Zurutuza U. and Tehrani S.N., Crowdroid: behaviour-based malware detection system for Android, 1st ACM workshop on Security and privacy in smartphones and mobile devices, 15-26 (2011)

[18] Polla M.L., Martinelli F., and Sgandurra D., A Survey on Security for Mobile Devices, Communications Surveys & Tutorials, IEEE, 15(1), 446–471 (2013)

[19] Tesfay W.B., Booth T., and Andersson K., Reputation Based Security Model for Android Applications, Trust, Security and Privacy in Computing and Communications, IEEE Computer Society, 896-901 (2012)

[20] Bing H., Analysis and Research of Systems Security Based on Android, Intelligent Computation Technology and Automation, 581–584 (2012)

[21] Bläsing T., Batyuk L., Schmidt A.D., Camtepe S.A. and Albayrak S., An Android Application Sandbox system for suspicious software detection, Malicious and Unwanted Software, 55-62 (2010)

[22] Vidas T., Votipka D. and Christin N., All Your Droid Are Belong To Us: A Survey of Current Android Attacks, The 5th USENIX conference on Offensive technologies, 10-10 (2011)

[23] Holla S. and Katti M.M., Android based mobile application development and its Security, International Journal of Computer Trends and Technology, 3(3), 486-490 (2012)

[24] Feth D., Pretschner A., Flexible Data-Driven Security for Android, The 2012 IEEE Sixth International Conference on Software Security and Reliability, 41-50 (2012)

[25] Portokalidis G., Homburg P., Anagnostakis K. and Bos H. Paranoid android: versatile protection for smartphones, Computer Security Applications Conference, 347–356(2010)

[26] Schmidt A.D. and Albayrak S., Malicious software for smartphones,https://www.dai- labor .de / fileadmin /files/publications/smart phone_ malware pdf (2008)

[27] Cheng J., Wong H.Y., Yang H. and Lu S., Smartsiren: virus detection and alert for smartphones, Mobile Systems,Applications, and Services, 258–271 (2007)

[28] Schmidt A.D., Bye R., Schmidt H.G., Clausen J., Kiraz O. and Yuksel K., A. Camtepe, and S. Albayrak, Static analysis of executables for collaborative malware detection on android, 2009 IEEE International Confernece on Communications, 1-5 (2009)

[29] Dini G., Martinelli F., Saracino A. and Sgandurra D.MADAM: a multi-level anomaly detector for android malware, http://www.iet.unipi.it/g.dini/research/papers/2012-MMM-ANCS.pdf (2012)

[30] Shabtai A., Kanonov U., Elovici Y., Glezer C. and Y. Weiss, Andromaly: a behavioural malware detection framework for android devices, Journal of Intelligent Information Systems, 38(1), 161-190 (2012).+