Volume 4, Issue 5, May 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Android Malware Detection Using SVM and GA

Mohini Tiwari, Ashish Kumar Srivastava, Nitesh Gupta
CSE Dept & NRI Institute of Information Science and Technology, Bhopal, Madhya Pradesh, INDIA

Abstract— Security is one of the main concerns for Smartphone users today. As the power and features of Smartphone's increase, so has their vulnerability for attacks by viruses etc. An Android OS could be attacked by hackers: Because it's Open platform, Users will access the Internet intensively and everyone can develop applications for Android. In previous technique described that how security can be improved of android operating system so that users can safely used the android smart phones, they have downloaded various android application from various android market on PC's and Decompiled those applications to fetch manifest file of downloaded apps and Applied various data mining techniques to find out permissions patterns in malware infected applications. So these techniques are not much more sufficient for android operating system. In this paper we are using research finding to identify malware in android devices not in the pcs which is actual benefits of research.

Keyword— Security Assessment, Software Security, Android

I. INTRODUCTION

However, there is a class of mobile devices that complies with the aforementioned characteristics. Smartphone's are high-end mobile devices that offer more advanced computing power and connectivity than a mobile phone. Because of that, they are becoming one of the principal ways people access social networks and they are closer to becoming "smart wallets" with the agreement among mobile operators on an approach to near field communications (NFC), which will enable mobile devices to act as "travel cards, money, tickets, and car keys. Both kinds of devices (smart phones and tablets) run lightweight operating systems (OS) that are optimized to run with limited energy consumption [16]. One of the most popular in the industry is Android.

A systematic characterization of existing Android malware ranging from their installation activation to the carried malicious payloads.

1.1 Malware Installation

By manually analyzing malware samples in our collection, we categorize existing ways Android malware use to install onto user phones and generalize them into three main social engineering-based techniques, i.e., (a).repackaging, (b).update attack, and (c).drive-by download. These techniques are not mutually exclusive as different variants of the same type may use different techniques to entice users for downloading.

(a) Repackaging

Repackaging is the mainly frequent techniques malware authors use to piggyback malicious payloads into popular applications or simply apps.In essence, malware authors may locate and download popular apps, enclose malicious payloads, and then re-assemble and submit the new apps to official and alternative Android sells. Clients could be susceptible by being attracted to download and install these infected apps.

(b) Update Attack

The first technique typically piggy- backs the entire malicious payloads into host apps, which could potentially expose their presence. The second technique makes it difficult for detection. Specifically, it may s Till repackage popular apps. But instead of enclosing the consignment as a complete, it only includes an renew component that will fetch or download the malicious payloads at runtime.

(c) Drive-by Download

The third technique applies the traditional drive-by download attacks to mobile freedom. Although they are not exploiting directly mobile browser vulnerabilities, they are really alluring users to download "interesting" or "feature-rich" apps. In our collection, we have four such malware families, i.e., GGTracker [16].

1.2 Security Issues faced by Android

Android is not much more secure because it seems, there are several security problems faced by the android. A number of them are mentioned below. 1. Android has no management over the apps being uploaded on its market. 2. Some apps exploit the services of another app while not creating a permission request. 3.Android's permission primarily based security model offers power is given to the user to create a decision whether or not an app ought to be trustworthy or not. This human part introduces lots of risk.4. The idea of an Open supply OS isn't solely open to legitimate developers however conjointly to hackers. so the complete framework of android can't be trustworthy once it involves building vital systems. 5. The android OS developers clearly state that they're not answerable for the protection of

secondary storage. 6. Any app on the android platform will access device information just like the GSM and SIM trafficker Ids while not the permission of the user age.

1.3 Support Vector Machine

Support Vector Machine (SVM), is one in every of best machine learning algorithms, that was projected in 1990's and used mostly for pattern recognition. This has additionally been applied to several pattern classification issues like image recognition, speech recognition, text categorization, face detection and faulty card detection, etc. Pattern recognition aims to classify information supported either a priori information or applied math statistical extracted from information, that may be a powerful tool in information separation in several disciplines. SVM may be a supervised kind of machine learning. rule during which, given a group of coaching examples, every marked as happiness to 1 of the various classes, associate SVM coaching rule builds a model that predicts the class of the new example. SVM has the larger ability to generalize the matter, which is that, the goal in applied math learning.

Related work

S. Powar and Dr. B. B. Meshram, surveyed on Android security framework, in this paper, Smartphone with open source operating systems are getting popular now days. Increased exposure of open source Smartphone is increasing the security risk also. This survey is about the current work done on the Android operating system. Some of the techniques, which can introduce a positive edge to the security area, are analyzed. These techniques are essentially to provide a better security and to make the Android security mechanism stretchier [1].

S. Kaur and M. Kaur presented review paper on implementing security on Android application. They described that how security can be improved of android operating system so that users can safely used the android smart phones [2].

S. Smalley and R. Craig presented the Android software stack for mobile devices defines and enforces its own security model for apps through its application-layer permissions model. In this paper, we motivate and describe our 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 demonstrate the benefits of our security enhancements for Android through a detailed analysis of how they mitigate a number of previously published exploits and vulnerabilities for Android [3].

W. Enck et. al introduce a study of Android application security. The concept of this paper is to better understand Smartphone application security by studying. They introduce the ded decompiler, and this decompiler improves Android application source code directly. They execute and design a parallel study of Smartphone applications based on static analysis of 21 million lines of recovered code [4].

W. Enck, et.al gives the concept about tracking System for real-time Privacy Monitoring on smartphones. Now a day smartphone operating systems frequently fail to provide users with adequate control over and visibility into how third-party applications use their personal information .They address these shortcomings with TaintDroid, an competent, system-wide dynamic contaminate tracking and analysis system capable of simultaneously tracking multiple sources of sensitive data [5].

Avik Chaudhuri presented and initiates a formal study of security on Android, they present a core typed language to describe Android applications, and to reason about their dataflow security properties. Their operational semantics and type system provide some necessary foundations to help both users and developers of Android applications deal with their security concerns [6].

Bernhard J. Berger et al. gives the concept about Software security has made great improvement, analysis tools are broadly used in industry for detecting common implementation-level security bugs. During the investigation they found some inconsistencies in the implementation of the Android security models. Based on the lessons learned they proposed several research topics in the area of reverse engineering that would support a security analyst during security assessments [7].

Machigar Ongtang et al. introduce the Smartphone concept. The security infrastructure available in current smartphone operating systems is largely underdeveloped. In this paper, they consider the security requirements of Smartphone applications and augment the existing Android operating system with a framework to meet them. They present Secure Application INTeraction (Saint), a modified infrastructure that governs install-time permission assignment [8].

Here author introduce the android framework, new software stack for mobile devices, key applications, system and middleware, this research provides a comprehensive security assessment of this framework and its security mechanisms the authors conducted a methodological qualitative risk analysis that identifies high-risk threats to the framework and any potential danger to information .they proposed several security solutions for mitigating these risks [9].

Here author show that the way in which permission based mechanisms are used on today's mobile platforms enables attacks by colluding applications that communicate over overt and hidden communication channels and these types of attacks permit applications to indirectly execute operations. They further show that on today's mobile platforms users are not made aware of possible implications of application collusion– quite the contrary–users are implicitly lead to believe that by approving the installation of each application separately, [10].

Adam Lackorzynski et al. gives the concept about Smartphone's became many people's primary means of communication. They presented a generic operating system framework that does away with the need for such hardware extensions. They encapsulate the original Smartphone operating system in a virtual machine. Their framework allows for highly secure applications to run side-by-side with the virtual machine [11].

David Barrera et al. introduce the Permission-based security models provide controlled access to various system resources. They present a methodology for the empirical analysis of permission-based security models. Their methodology is of independent interest for visualization of permission based systems beyond our present Android-specific experiential analysis. They offer some discussion identifying potential points of improvement for the Android acquiescence model, trying to increase expressiveness where needed without increasing the total number of permissions or overall complexity [12].

Here author capitalize on earlier approaches for dynamic analysis of application behavior as a means for detecting malware in the Android platform. The method is shown to be an effective means of separating the malware and alerting the users of a downloaded malware. They showed the potential for avoiding the spreading of a detected malware to a larger community [13]. Suhas Holla and Mahima M Katti discussed on Android mobile platform for the mobile application development. They discuss a layered approach for android application development where they can develop application which downloads data from the server. Also an Android Application Sandbox (AAS and box) which is able to perform both static and dynamic analysis on Android programs to automatically detect suspicious applications is also discussed [14].

Here author describe MADAM, a Multi-level Anomaly Detector for Android Malware. MADAM parallel examines Android at the kernel-level and user-level to detect real malware infections using machine learning techniques to distinguish between malicious ones and standard behaviors. The first prototype of MADAM is able to detect several real malware found in the wild. The device usability is not accepted by MADAM [15].

II. PROBLEM STATEMENT

Present malware detection applications based on signature based detection of malwares in android. In previous work there, have not developed any application (android application) to detect malware. They have downloaded various android application from various android market on PC's and Decompiled those applications to fetch manifest file of downloaded apps and Applied various data mining techniques to find out permissions patterns in malware infected applications. So these techniques are not much more sufficient and user friendly for android operating system.

III. PROPOSED SOLUTION

Now we proposed Permission based android malware detection using Genetic Algorithm. We are using research finding to identify malware in android devices not in the pcs which is actual benefits for research. This new concept will have sample good ware and malware training sample and GA will update training sample. K-mean classification with GA is used to malware detection in android. This application also detects malwares in the installed applications in android. This application not required network to detect malware as cloud based application required.

IV. PROPOSED ALGORITHM

Algorithm 01: For Classifier Module
Prepare the initial training samples of permissions binary pattern (PBP) of good apps and malwares infected apps
Prepare array with mark the +1 for good ware apps and -1 for malware apps
Train the SVM using PBP's and class array to generate SVM classifier
Now SVM classifier can identify the malware if new PBP is supplied to it output will be +1 or -1
Output +1 means good ware and -1 means Malware
Algorithm 02: For Genetic Algorithm module
Retrieve the PBP's of all installed apps in the device
Check for any new PBP which is not in the training samples of SVM
Pass the identified new PBP's and some of good ware and malware PBP's to Genetic Algorithm (GA)
Now GA will generate optimized PBP's with its class (0) or (1)
Change class 0 to -1 of generated PBP's
Update the SVM Training Samples
Again Train SVM with new Training Samples to generate updated SVM Classifier
Algorithm 03: For Detector Module
Retrieve the list of installed apps in the device

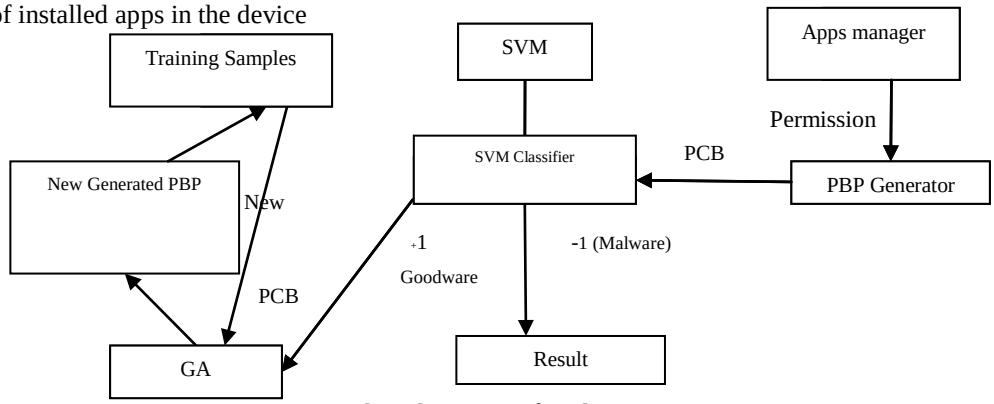

Figure1: Proposed Architecture of Malware Detection

Check the PBPs using SVM classifier (which is generated using algorithm 01) to detect malwares
Get the permission requested by each app
Prepare the permissions binary pattern (PBP) for each apps
If new PBP's found then Pass them to GA module (algorithm 02).

V. RESULT ANALYSIS

Permission set requested by well known malwares are as follows

Permissions / Malwares	**GDD**	**DCC**	**Pas**	**JR**	**GD**
ACCESS COARSE LOCATION	1	0	0	0	0	0	1
ACCESS FINE LOCATION	1	0	0	0	0	0	1
ACCESS NETWORK STATE	0	0	1	0	1	0	1
ACCESS WIFI STATE	0	1	0	0	1	0	1
BROADCAST PACKAGE REMOVED	0	0	0	0	1	0	0
CALL PHONE	1	0	0	0	0	0	1
CHANGE WIFI STATE	0	1	0	0	0	0	0
DELETE PACKAGES	0	0	0	0	0	0	1
DEVICE POWER	0	0	0	0	1	0	0
INSTALL PACKAGES	0	0	0	0	0	0	1
INTERNET	1	1	1	0	1	1	1
KILL BACKGROUND PROCESSES	0	0	0	0	1	0	0
MOUNT UNMOUNT FILESYSTEMS	1	0	0	0	0	0	0
PROCESS OUTGOING CALLS	0	0	0	0	0	0	1
READ CONTACTS	1	0	0	0	0	0	0
READ PHONE STATE	1	1	1	0	1	0	1
READ SMS	0	0	0	0	1	0	1
RECEIVE BOOT COMPLETED	0	0	0	0	0	0	1
RECEIVE SMS	0	0	0	1	1	1	1
SEND SMS	1	0	0	1	1	1	1
SET WALLPAPER	1	0	0	0	0	0	0
WAKE LOCK	0	0	0	0	1	0	0
WRITE APN SETTINGS	0	0	0	0	1	0	0
WRITE CONTACTS	1	0	0	0	0	0	0
WRITE EXTERNAL STORAGE	1	0	0	1	1	0	1
WRITE SMS	0	0	0	0	1	0	0

Note: 1 indicates permission requested and 0 indicates not requested by particular malware. Where:

G= Geinimi
DD = DroidDream
CC = CounterClank
P = Pjapps
as = asSMS
JR = Jimm Russia
GD = Gold Dream

Permission table is for reference only for some of malware application permissions. Previous technique results was generated using data mining on the thousands of android application using de-compilation and WEKA tools and our result will be generated on Android device. Malware detector application scans the manifest file of installed applications in smartphone or tablet to know permission set of each one. These permissions are transforms into binary pattern using above mentioned table and passed the SVM to identify malware.

VI. CONCLUSION

Android is one of the most popular open source operating system for mobile platforms. Android provide a basic set of permissions to protect phone resources. But still the security area is underdeveloped. Successful attack on Androids may: Expose private information, Prevent T-Mobile customers from using T-Mobile services, Flood T-Mobile's customer service infrastructure and personnel. No easy way exists to "fix" mobile devices and especially Android. .In this paper we introduce security issues in the Android based Smartphone. The integration of technologies into an application certification process requires overcoming logistical and technical challenges. Android gives additional security than other mobile phone platforms. The platform must offer an application environment that ensures a complete solution for all security concern i.e. Prevent privilege escalation by application, Prevent data leakage by apps. Prevent bypass of security features, inflict legal restrictions on data, Protect reliability of apps and data, Beneficial for consumers and businesses.

REFERENCES

[1] S. POWAR, DR. B. B. MESHRAM, "SURVEY ON ANDROID SECURITY FRAMEWORK", INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH AND APPLICATIONS (IJERA) ISSN: 2248-9622 VOLUME 3, ISSUE 2, MARCH -APRIL 2013.

[2] S. KAUR AND M. KAUR, "REVIEW PAPER ON IMPLEMENTING SECURITY ON ANDROID APPLICATION", JOURNAL OF ENVIRONMENTAL SCIENCE, COMPUTER SCIENCE AND ENGINEERING & TECHNOLOGY (JECET), VOLUME 2, NO. 3, JUNE-AUGUST 2013.

[3] S. SMALLEY AND R. CRAIG, "SECURITY ENHANCED (SE) ANDROID: BRINGING FLEXIBLE MAC TO ANDROID", TRUSTED SYSTEMS RESEARCH NATIONAL SECURITY AGENCY2005.

[4] W. ENCK, D. OCTEAU, P. MCDANIEL AND S. CHAUDHURI "A STUDY OF ANDROID APPLICATION SECURITY", SYSTEMS AND INTERNET INFRASTRUCTURE SECURITY LABORATORY, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, THE PENNSYLVANIA STATE UNIVERSITY. IN PROCEEDINGS OF THE 20TH USENIX CONFERENCE ON SECURITY, P.21-21, AUGUST 08-12, SAN FRANCISCO, CA, 2011.

[5] W. ENCK, P. GILBERT, B. G. CHUN, L. P. COX, J. JUNG, P. MCDANIEL AND A. N. SHETH, "TAINTDROID: AN INFORMATION-FLOW TRACKING SYSTEM FOR REALTIME PRIVACY MONITORING ON SMARTPHONES", 9TH USENIX SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION (OSDI'10).

[6] AVIK CHAUDHURI, UNIVERCITY OF MARYLAND AT COLLEGE PARK: "LANGUAGE-BASED SECURITY ON ANDROID".

[7] BERNHARD J. BERGER, MICHAELA BUNKE, AND KARSTEN SOHR STUDY ON "AN ANDROID SECURITY CASE STUDY WITH BAUHAUS". CENTER FOR COMPUTING TECHNOLOGIES (TZI), UNIVERSIT¨AT BREMEN, GERMANY. IN THE PROCEEDINGS OF 2011 18TH WORKING CONFERENCE ON REVERSE ENGINEERING (WCRE), LIMERICK, OCTOBER 17-20, 2011.

[8] MACHIGAR ONGTANG, STEPHEN MCLAUGHLIN, WILLIAM ENCK AND PATRICK MCDANIEL "SEMANTICALLY RICH APPLICATION-CENTRIC SECURITY IN ANDROID". DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING THE PENNSYLVANIA STATE UNIVERSITY, UNIVERSITY PARK, PA 16802. IN PROCEEDINGS OF THE ANNUAL COMPUTER SECURITY APPLICATIONS CONFERENCE (ACSAC '09), AUSTIN, TX, USA, DECEMBER 6–10, 2009.

[9] A. SHABTAI, Y. FLEDEL, U. KANONOV, Y. ELOVICI, S. DOLEV, AND C. GLEZER. GOOGLE ANDROID: A COMPREHENSIVE SECURITY ASSESSMENT. SECURITY PRIVACY, IEEE, 8(2):35 –44, MARCH-APRIL 2010.

[10] CLAUDIO MARFORIO, AUR´ELIEN FRANCILLON, SRDJAN CAPKUN "APPLICATION COLLUSION ATTACK ON THE PERMISSION-BASED SECURITY MODEL AND ITS IMPLICATIONS FOR MODERN SMARTPHONE SYSTEMS", DEPARTMENT OF COMPUTER SCIENCE, ETH ZURICH, SWITZERLAND.

[11] ADAM LACKORZYNSKI, MATTHIAS LANGE, ALEXANDER WARG, STEFFEN LIEBERGELD, MICHAEL PETER "L4ANDROID: A GENERIC OPERATING SYSTEM FRAMEWORK FOR SECURE

[12] DAVID BARRERA, H. GÜNE¸S KAYACIK, P.C. VAN OORSCHOT, ANIL SOMAYAJI "A METHODOLOGY FOR EMPIRICAL ANALYSIS OF PERMISSION-BASED SECURITY MODELS AND ITS APPLICATION TO ANDROID" , IN PROCEEDINGS OF THE 17TH ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, CCS '10, PAGES 73–84, NEW YORK, NY, USA, ACM. SCHOOL OF COMPUTER SCIENCE, CARLETON UNIVERSITY OTTAWA, ON, CANADA 2010.

[13] IKER BURGUERA,URKO ZURUTUZA, SIMIN NADJM-TEHRANI "CROWDROID: BEHAVIOR-BASED MALWARE DETECTION SYSTEM FOR ANDROID".INPROCEEDINGS OF THE 1ST ACM WORKSHOP ON SECURITY AND PRIVACY IN SMARTPHONES

AND MOBILE DEVICES (SPSM'11),, PAGES 15-26, 2011, ACM NEW YORK, NY, USA ©2011. CHICAGO, IL, USA, OCTOBER 17, 2011.

[14] SUHAS HOLLA, MAHIMA M KATTI. "ANDROID BASED MOBILE APPLICATION DEVELOPMENT AND ITS SECURITY", DEPARTMENT OF INFORMATION SCIENCE & ENGG, R V COLLEGE OF ENGINEERING BANGALORE, INDIA. INTERNATIONAL JOURNAL OF COMPUTER TRENDS AND TECHNOLOGY- VOLUME3ISSUE3- 2012.

[15] GIANLUCA DINI, FABIO MARTINELLI, ANDREA SARACINO, AND DANIELE SGANDURRA. "MADAM: A MULTI-LEVEL ANOMALY DETECTOR FOR ANDROID MALWARE" .IN PROCEEDING MMM-ACNS'12 PROCEEDINGS OF THE 6TH INTERNATIONAL CONFERENCE ON MATHEMATICAL METHODS, MODELS AND ARCHITECTURES FOR COMPUTER NETWORK SECURITY: COMPUTER NETWORK SECURITY PAGES 240-253, 2012.

[16] HTTP://WWW.SCRIBD.COM/DOC/219864151/ATTACKS-ON-ANDROID.

