ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

Review of Behavior Malware Analysis for Android

Nitin Padriya, Nilay Mistry

Abstract*— Android based Smartphone are now a day's getting more popularity. With the use of Smartphone user must always concern about the security breaching and malicious attacks. Here we introduce an approach for proactive malware detection working by abstraction of program behaviors. Suspicious behaviors are detected by comparing trace abstractions to reference malicious behaviors. The sensitive power of concept allows us to grip common mistrustful behaviors rather than specific malware code and then, to distinguish malware transformation [18].We present and discuss an implementation validating our approach. First have to analyze the programs or apps, then represented them as trace languages, which are abstracted by altering with respect to elementary behavior patterns, defined as regular string rephrasing systems. This paper review the state of the art on threats, vulnerabilities , We aimed at existing approaches to protecting mobile devices against these classes of attacks into different categories, based upon the detection principles, architectures, collected data and operating systems, especially focusing on IDS-based models and tools.

***Index Terms*— Malware, Security, Android.**

I. INTRODUCTION

Mobile devices, such as smart phone or PDA have become more and more widespread, and often essential in our everyday life. Usually they contain lots of sensitive information, like a list of contacts, ingoing/outgoing call, text messages, and on latest model a calendar of our schedule, emails, our current position (if the phone has embedded GPS). Latest models feature a complete OS, but for many people they are just phones, so there is an under-estimation of the risk connected to phone security. This makes. Phones an interesting target for malicious users. Damages that a user can sustain are financial loss, privacy and confidentiality, slowdown of processing speed, battery life. Reverse Engineering of Malware Analysis is a process which is used by forensic investigators and system engineers to analyses the flow, operation, functionality of malware using reverse engineering tools. This is useful in understanding the files, services, code and parameters which were added or modified by the malicious software. It includes analysis in two phases namely Static (Behavioral) Analysis and Dynamic (Code) Analysis.

II. THREAT MODEL

The mobile threat model includes three types of threats: Malware, grayware, and personal spyware. We distinguish between the three based on their delivery method, legality, and notice to the user. This paper focuses specifically on malware; personal spyware and grayware use different attack vectors, have different motivations, and require different defense mechanisms. [3]

Malware. Malware gains access to a device for the purpose of stealing data, damaging the device, or annoying the user, etc. The attacker defrauds the user into installing the malicious application or gains unauthorized remote access by taking advantage of device vulnerability. Malware pro- vides no legal notice to the affected user. This threat includes Trojans, worms, botnets, and viruses.

Personal Spyware. Spyware collects personal information such as location or text message history over a period of time. With personal spyware, the attacker has physical access to the device and installs the software without the user's knowledge. Personal spyware sends the victim's information to the person who installed the application onto the victim's device, rather than to the author of the application. Personal spyware is honest about its purpose to the person who purchases and installs the application. How- ever, it may be illegal to install personal spyware on another person's smart phone without his or her authorization. [3]

Grayware. Some legitimate applications collect user data for the purpose of marketing or user providing. Grayware Spies on users, but the companies that distribute grayware do not aim to harm users. Pieces of grayware provide real Functionality and value to the users. The companies that distribute grayware may disclose their collection habits in their privacy policies, with varying degrees of clarity. Gray ware sits at the edge of legality; its behavior may be legal or illegal depending on the jurisdiction of the complaint and the wording of its privacy policy. [3]

III. MALWARE ANALYSIS

A. Basics Of Malware Analysis

Malwares are evolving in a rapid manner and combat measures to stop them have become difficult because they use new signatures, encapsulation which prevents it from being detected. Anti-Virus products have been releasing daily updates which detect almost all the attacks, some of them narrowly escape. It is essential that a reverse engineer must analyze such malwares which change the registry values, tamper data, and download payloads in short which shows unusual behavior. Reverse Engineer must analyze malware of that particular Operating system and study the environmental variables and activity performed by that malicious software.

B. Types of Malware Analysis

- **Static Analysis** also called behavioral analysis, which is used to analyze and study the behavior of malware. I can study how malware interact with the environment, services

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

added, files tampered, data captured, network connection made, port opening etc. [4] Collected data is reconstructed and mapped together to get a complete picture of the analysis.

- **Dynamic Analysis** also called code analysis, which is used to analyze the code of the malicious software. As it is very difficult to get the source code of the malware especially executables we need to analyze the binary code. There are debuggers and decompiles which are used to convert the malware to its binary form or assembly level. By analyzing the code, a reverse engineer will come to know the exact malicious code which is embedded in the actual code.

C. Components of Malware Analysis

Malware creators use different techniques to develop malicious software so it is difficult to specify a common factor in all the malwares. Each malware have a different signature, programming language and a packer. [4] Around 500 packers are released which are used by the attackers in order to prevent the malware from being detected by the anti-virus applications. Packers are used so that the code is compressed using tools like 7-zip or Win Rar etc. Compressed code will be harder to detect as there will be a session with less CPU Utilization unlike larger code which utilizes maximum CPU utilization and sessions. Another method is to add more protection by using encryption to prevent from being detected. Even though anti-virus will unpack the malware they will just scan the encrypted version of the code Malware may be such as Net cat, VNC, Exposits, Script, Spyware, and Adware Etc. Thus malware may contain multiple malicious components which ensure that at least one of them will cause damage to the system

IV. MALWARE-DETECTION

Current commercial anti-malware tools are constantly challenged by the increased frequency of malware outbreaks. Several malware analysis and detection approaches have been proposed to minimize the distribution of malicious programs. However, malware writers deploy new techniques such as obfuscation and altering program behavior [13] in order to create new, undetectable, malicious programs that evade state-of-the-art detectors. I provide, through selective reference to some of the literature, a clearer understanding of the existing malware Detection techniques. Malware detection approaches presented in the literature are based on various analysis strategies that are common in computer software analysis, i.e. static, dynamic and hybrid. Moreover, we propose classifying existing malware detection approaches into five broad categories:

- Signature-based
- Behavior-based
- Heuristic-based
- Model checking-based
- Semantics-based

Before reviewing the development of malware detection research, we categorized it into three tiers. Research into the detection approach is placed at the top level of the hierarchy, followed by research into input representations and analysis types.

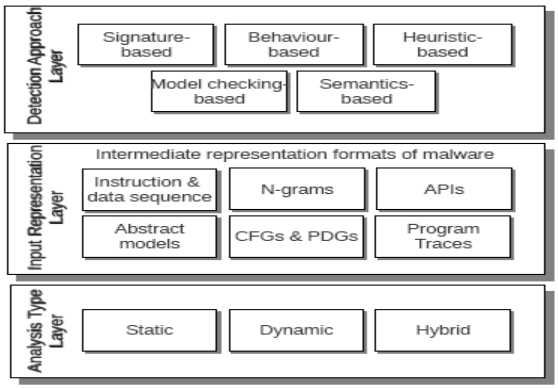

Fig 1. A Three-Tier Hierarchy Diagram of Malware-Detection Research

Input representations are the intermediate representation formats of malware programs Produced as inputs to the malware detectors. The analysis type divides Detection techniques into three groups: static, dynamic and hybrid. Figure 1.Shows our three-tier hierarchy for malware-detection research. Our analysis in this chapter of the existing malware detection research is based on this hierarchy.

V. TOOLS OF MALWARE ANALYSIS

Various tools are selected by the reverse engineer for the behavioral and code analysis. The following is a list of tools used for reverse engineering malware by most reverse engineers.

- Creation of Lab -Virtual Box, VMware, Sandbox GFI.
- Static Analysis - Process Monitor, Wire shark, PEiD, TCPView, WinHex, ProcessExplorer, Win analysis, Strings.
- Dynamic Analysis -Ollydbg, IDA Pro, Dex2jar, JD-GUI, Baksmali, Apktool.

ACKNOWLEDGEMENT

The challenges and issue of analysis of such malwares is that attackers use different programming languages, packers to hide the code more over use encryption and obfuscation techniques to prevent from getting the source code. Tools for Reverse Engineering needs to be advanced as they lack exposure over many programming languages and platforms etc.

REFERENCES

[1] MARIANTONIETTA La Polla, Fabio Martinelli, and Daniele Sgandurra, A Survey on Security for Mobile Devices" IEEE COMMUNICATIONS SURVEYS & TUTORIALS, ACCEPTED FOR PUBLICATION" 1553-877X/12/ 2012 IEEE.

[2] Addodil jo elv arg hese, Pro. F STUART WALKER, Dissecting Andro Malware, 2011 The SANS Institute.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

[3] ADRIENNE Porter Felt, Matthew Finifter, Erika Chin,Steven Hanna, and David Wagner,A Survey of Mobile Malware in the Wild,ACM 78-1-4503-1000/11/10,October17,2011,Chicago.

[4] Rajdeep Chakraborty, "Detailed analysis of the continuously evolving threat of Malwares", Retrieved http://www.malwareinfo.org/library/whitepapers/MalwareAnalysisHow2.pdf,Last Accessed: 24 August, 2011.

[5] Dennis Distler, "Malware Analysis: A Introduction",RetrievedFrom:http://www.sans.org/reading_room/whitepapers/malicious/malware-analysis-introduction_2103.

[6] Stephen. A.Ridley, "Android Malware Reverse Engineering",Retrievedfrom:http://dl.dropbox.com/u/2595211/HelloMotoAndroidReversing.

[7] Google Android, Retrieved http://developer. android.com/guide/basics/what-is-android.html.

[8] Troy Vennon, "Threat Analysis of the Android Market",http://www.globalthreatcenter.com/wp-content/uploads/2010/06/Android-Market-Threat-Analysis-6-22-10-v1.pdf, Last Accessed: 24 August, 2011.

[9] Johannes Kinder, Stefan Katzenbeisser, Christian Schallhart, and Helmut Veith. Proactive detection of computer worms using model checking. IEEE Transactions on Dependable and Secure Computing, 7:424{438, October 2010.

[10] Dong-Jie Wu1, Ching-Hao Mao2 "DroidMat: Android Malware Detection through Manifest and API Calls Tracing"2012 Seventh Asia Joint Conference on Information Security. 978-0-7695-4776-3/12 /IEEE.

[11] Takamasa Isohara, Keisuke Takemori and Ayumu Kubota" Kernel-based Behavior Analysis for Android Malware Detection" 2011 Seventh International Conference on Computational Intelligence and Security 978-0-7695-4584-4/11 2011 IEEE.

[12] National Institute of Standards and technology, from:http://csrc.nist.gov/publications/nistpubs/800-83/SP800-83.pdf, Last Accessed: 24 August, 2011.

[13] Emre Erturk" A Case Study in Open Source Software Security and Privacy: Android Adware" World Congress on Internet Security (WorldCIS-2012). 978-1-908320-04/9/$25.00©2012 IEEE.

[14] Patrik Lantz" Analysis of Malicious and Benign Android Applications" 2012 32nd International Conference on Distributed Computing Systems Workshops. 1545-0678/12 / 2012 IEEE.

[15] Anthony Desnos " Android : Static Analysis Using Similarity Distance" 2012 45th Hawaii International Conference on System Sciences. 978-0-7695-4525-7/12 2012 IEEE.

[16] Thomas Bl¨asing, Leonid Batyuk, Aubrey-Derrick Schmidt, Seyit Ahmet Camtepe, and ahin Albayrak" An Android Application Sandbox System for Suspicious Software Detection" 978-1-4244-9356-2/10/c_2010 IEEE.

[17] Philippe Beaucamps, Isabelle Gnaedig, Jean-Yves Marion, "Behavior Abstraction in Malware Analysis" 1st International Conference on Runtime Verification 6418 (2010) 168-182.

AUTHOR PROFILE

Nitin C. Padriya born in Gujarat, India in June, 1984. He received his bachelors of Engineering in Information Technology from Gujarat University, India in June 2007.And is currently pursuing his Master of Engineering in Computer Engineering(Wireless Mobile Computing) from Gujarat Technological University (India).His research interests include Wireless Grid Computing, Mobile Computing.

Nilay R. Mistry born in Gujarat, India in December 27, 1986. He received his bachelors of Engineering in Computer from Gujarat University, India in June 2009. And Master in Computer Science and Engineering from NIRMA University in June 2011, Gujarat. His research interests include digital forensics and cyber security.