A Novel Approch to Defeat Split Personality Malware

Nisha Lalwani,
Department of Computer Science and Engineering
Shri Ramdeobaba College of Engineering and Management, Katol Road, Nagpur-13, India

M.B. Chandak,
Department of Computer Science and Engineering
Shri Ramdeobaba College of Engineering and Management, Katol Road, Nagpur-13, India

Abstract:
Malware, also known as malicious code and malicious software, refers to a program that is inserted into a system, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system or otherwise annoying or disrupting the victim. Security analysts extensively use virtual machines to analyze sample programs and study them to determine if they contain any malware. Malware detection is again a crucial aspect of software security. This paper is intended to help organizations understand the threats posed by malware and mitigate the risks associated with malware incidents. In addition to providing background information on the major categories of malware, this paper majorly focuses on Analysis Aware Malware also called as Split Personality Malware which checks for the presence of Malware Analysis tools and behaves in a benign manner thus escaping detection.

INTRODUCTION

Impelled by the proliferation of high speed connections and the global coverage, Internet has become a powerful means for knowledge sharing as well as commercialization. The increasing dependence on the Internet, however, also makes it an obvious target for the miscreants to spread computer viruses and other types of malicious software (Malware). The power of malware has reached the level where it can not only penetrate, manipulate and destroy information systems but can even reside on them indefinitely gaining complete control over them without the user getting the slightest hint. This is mainly due to two important factors[5]. Firstly, plentiful vulnerabilities in the operating systems, browsers and other applications are being continuously discovered and exploited by the malware developers prior to any patches being developed against them by the security researchers. Such attacks are termed as zero day attacks. Secondly, the malware developers make use of several obfuscation techniques in order to evade detection by the various anti-malware products especially the ones based on signature detection schemes. There are daily reports in the technical and popular press about new vulnerabilities and new types of attacks, and the rapidly increasing economic incentives are sure to catalyze this activity for a long time to come. Well known examples of malicious activity include denial of service, spam, information gathering, and resource gathering. Studies reveal that the impact of Malware is getting worse day by day and that its overwhelming number makes it a Worldwide epidemic!

This paper is divided into 5 sections. The first section provides an overview of the various categories of malware, which include viruses, worms, Trojan horses, and malicious mobile code, as well as combinations of these, known as blended attacks. Section 2 focuses on Literature Survey done for Split Personality malware. In section 3 we have described various techniques used for VM Detection. In section 4 we propose our approach to detect and defeat Split Personality malware. In section 5, we conclude.

I. CATEGORIES OF MALWARE

Malware has become the greatest external threat to most systems, causing damage and requiring extensive recovery efforts within most organizations. Malware is divided into the following major categories:

1. Viruses:
A virus self-replicates by inserting copies of itself into host programs or data files. Viruses are often triggered through user interaction, such as opening a file or running a program. Viruses can be divided into the following two subcategories:

Compiled Viruses: A compiled virus is executed by an operating system. Types of compiled viruses include file infector viruses, which attach themselves to executable programs; boot sector viruses, which infect the master boot records of hard drives or the boot sectors of removable media; and multipartite viruses, which combine the characteristics of file infector and boot sector viruses.

Interpreted Viruses: Interpreted viruses are executed by an application. Within this subcategory, macro viruses take advantage of the capabilities of applications. macro programming language to infect application documents and document templates, while scripting viruses infect scripts that are understood by scripting languages processed by services on the OS.

Proc. of the Intl. Conf. on Advances in Computer, Electronics and Electrical Engineering
Editor In Chief Dr. R. K. Singh. ISBN: 978-981-07-1847-3

2. Worms:
A worm is a self-replicating, self-contained program that usually executes itself without user intervention. Worms are divided into two categories:

Network Service Worms: A network service worm takes advantage of a vulnerability in a network service to propagate itself and infect other systems.

Mass Mailing Worms: A mass mailing worm is similar to an e-mail borne virus but is self-contained, rather than infecting an existing file.

3. Trojan Horses:
A Trojan horse is a self-contained, non-replicating program that, while appearing to be benign, actually has a hidden malicious purpose. Trojan horses either replace existing files with malicious versions or add new malicious files to systems. They often deliver other attacker tools to systems.

4. Malicious Mobile Code:
Malicious mobile code is software with malicious intent that is transmitted from a remote system to a local system and then executed on the local system, typically without the user's explicit instruction[3]. Popular languages for malicious mobile code include Java, ActiveX, JavaScript, and VBScript.

5. Blended Attacks:
A blended attack uses multiple infection or transmission methods. For example, a blended attack could combine the propagation methods of viruses and worms.

6. Tracking Cookies:
A tracking cookie is a persistent cookie that is accessed by many Web sites, allowing a third party to create a profile of a user.s behavior. Tracking cookies are often used in conjunction with Web bugs, which are tiny graphics on Web sites that are referenced within the HTML content of a Web page or e-mail. The only purpose of the graphic is to collect information about the user viewing the content.

7. Attacker Tools:
Various types of attacker tools might be delivered to a system as part of a malware infection or other system compromise. These tools allow attackers to have unauthorized access to or use of infected systems and their data, or to launch additional attacks. Popular types of attacker tools are as follows:

Backdoors: A backdoor is a malicious program that listens for commands on a certain TCP or UDP port. Most backdoors allow an attacker to perform a certain set of actions on a system, such as acquiring passwords or executing arbitrary commands. Types of backdoors include zombies (also known as bots), which are installed on a system to cause it to attack other systems, and remote administration tools, which are installed on a system to enable a remote attacker to gain access to the system's functions and data as needed.

Keystroke Loggers: A keystroke logger monitors and records keyboard use. Some require the attacker to retrieve the data from the system, whereas other loggers actively transfer the data to another system through e-mail, file transfer, or other means.

Rootkits: A rootkit is a collection of files that is installed on a system to alter its standard functionality in a malicious and stealthy way. A rootkit typically makes many changes to a system to hide the rootkit's existence, making it very difficult to determine that the rootkit is present and to identify what the rootkit has changed.

Web Browser Plug-Ins: A Web browser plug-in provides a way for certain types of content to be displayed or executed through a Web browser. Attackers often create malicious Web browser plug-ins that act as spyware and monitor all use of the browser.

Split Personality Malware: To thwart automated screening Malware authors have developed a number of ways to check for the presence of Malware analysis tools and popular sandbox environment. When the malware detects the presence of Malware analysis system, it typically suppresses the execution of malicious functionality or simply terminates. This kind of Malware is also known as Analysis Aware Malware.

E-Mail Generators: An e-mail generating program can be used to create and send large quantities of e-mail, such as malware, spyware, and spam, to other systems without the user's permission or knowledge.

Attacker Toolkits: Many attackers use toolkits containing several different types of utilities and scripts that can be used to probe and attack systems, such as packet sniffers, port scanners, vulnerability scanners, password crackers, remote login programs, and attack programs and scripts.

In addition to malware, there are also a few common non-malware threats that are often associated with malware. Phishing uses computer-based means to trick users into revealing financial information and other sensitive data. Phishing attacks frequently place malware or attacker tools on systems. An additional malicious content threat is virus hoaxes false warnings of new malware threats.

II. LITERATURE SURVEY

During the phase of our literature survey, we noted two key factors responsible for the alarming, uncontrolled growth of malware. The reasons are listed as follows:

1. Increase in the number of Zero Day attacks.
2. Development and eventual growth of Analysis Aware/Split Personality Malware.

The above two factors are correlated. Zero day attacks are attacks carried out by exploiting the unknown vulnerabilities in operating systems, browsers and other user applications; against which no securities patches are available. And the Analysis Aware Malware also known as Split Personality Malware, are malware that prevent these vulnerabilities from being known to the security researchers. This class of malware adopts an obfuscation technique where in they hide their malicious intent on discovering that that they are under analysis. In other words, they incorporate the ability to detect the presence of malware analysis tools and on detection of any of such tools; they behave like regular harmless binaries effectively shielding the unknown exploits that they carry [6].

A. VIRTUALIZATION, SECURITY RESEARCH & MALWARE

Malware analysts use a wide variety of tools to carry out the analysis. Virtualization has emerged as a very useful technology in the field of security research and has gained widespread acceptance in the fraternity. It is very popular amongst malware researchers since they can intrepidly execute suspicious malware samples in virtual machines without having their systems affected [4]. Since many malware tend to destabilize the host systems, allowing them to run in a virtual environment increases the productivity of the analysts. This decreases the time and cost that the analysts need to study malware behaviors enabling them to build patches against the vulnerabilities that the malware exploit allowing them to run in a virtual environment increases the productivity of the analysts[7]. This decreases the time and cost that the analysts need to study malware behaviours enabling them to build patches against the vulnerabilities that the malware exploit.

B. ANALYSIS AWARE MALWARE

With security researchers relying on Virtual Machine Environment (VME), debuggers and sandboxes in their analysis work, attackers and their malicious codes have a significant stake in detecting the presence of these malware analysis tools. Virtualization, by its very nature, creates systems that have different characteristics from the real machines. From a theoretical perspective, any difference between the virtual and the real could lead to a fingerprinting opportunity for attackers. Thus, Malware writers have developed a new class of malware called Analysis Aware Malware or Split Personality Malware[6]. This class detects the presence of malware analysis tools such as Virtual Machines (VM), debuggers and sandboxes and then either terminates execution or hides its malicious nature by executing like a benign application. As a result, the malware escapes detection from a casual malware analyst.

However, the malware developers have once again upped the ante by adding analysis awareness functionality into their malware. They detect the presence of malware analysis tools such as Virtual Machines (VM), debuggers and sandboxes and then either terminate execution or hide their malicious nature by executing like a benign application. As a result, they escape detection from a casual malware analyst. This category of malware is known as Analysis Aware malware or Split Personality malware[7].

III. VM DETECTION TECHNIQUES

There are various ways of VM detection, all of which can be classified in one of the following categories:

A. Hardware Fingerprinting

Hardware Fingerprinting involves looking for specific virtualized hardware [8]. It can reveal a plethora of information about VM specific components required for reliable detection.

B. Registry Check

The registry entries contain hundreds of references to the string "VMware" in the guest OS. Checking the registry values for certain keys clearly reveals the VM presence [8]. The following are a few examples:

HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\Scsi\Scsi Port1\Scsi Bus 0\Target Id 0\Logical Unit Id 0\Identifier
- ➔ VMware, VMware Virtual S1.0

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Class\{4D36E968-E325-11CE-BFC1-08002BE10318}\0000\DriverDesc
- ➔ VMware SCSI Controller

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Class\{4D36E968-E325-11CE-BFC1-08002BE10318}\0000\ProviderName
- ➔ VMware, Inc.

C. Memory Check

This technique involves looking at the values of specific memory locations after the execution of instructions such as SIDT (Store Interrupt Descriptor Table), SLDT (Store Local Descriptor Table), SGDT (Store Global Descriptor Table), and STR (Store Task Register) [9]-[12]. It is the most widespread detection

technique employed by the present day VM detecting malware.

D. VM Communication Channel Check

This check involves detecting the presence of a host-guest communication channel. The IN instruction is a privileged instruction which when executed from ring 3 of a protected mode OS such as Windows, raises the exception "EXCEPTION PRIV INSTRUCTION"' [8]. However, when it is running on VMware, no such exception is generated. Instead, VMware initiates guest to host communication by calling the „IN instruction. If the magic number („VMXh) is returned to the register EBX, then it is certain that the program is running inside VMware.

E. Timing Analysis

An obvious yet rare attack against a Virtual Machine is to check a local time source, such as the "Time Stamp Counter" (TSC). We briefly restate the concept behind this attack discussed in a previous work [7]. Translation Lookaside Buffers (TLBs) can be explicitly flushed out and then the time to access a new page is determined by reading the TSC before and after the access. This duration can be averaged out over the number of TLBs to be filled. Next, the TLBs are filled with known data by accessing a set of present pages and the time to access a cashed page is determined as before. This value can also be averaged over the number of pages in the TLBs. Now, the CPUID instruction is executed. CPUID is the only VM sensitive instruction which on execution flushes out at least some of the TLBs as a side effect. Now each of the pages that were present in the VM is accessed again. If any of the page's access time matches that of a new page, the presence of a VM is revealed!

F. Process & File Check

There are many VMware specific processes such as VMwareUser.exe, vmacthlp.exe, VMwareService.exe, VMwareTray.exe that constantly run in the background. There also exist some VMware specific files and folders [8]. Hence querying for these objects could also serve as a method for VM detection. Though this method could easily be fooled, when combined with other detection techniques, it could obtain more reliable results.

IV. OUR APPROACH

Our main aim is to tackle this class of malware. Current efforts mainly focus on flagging the Split Personality malware and once flagged they resort to analyzing them on a native machine to bring out their malicious nature. We aim to develop a solution to counter Split Personality Malware that use VM detection Logic. The countering mechanism should not only enable the detection of this class of Malware but also trick them into believing that they are running on a host machine even when they are actually running on VM. This is done in order to bring out the malicious nature of the malware for the purpose of effective malware analysis in virtualized environment.

The main objective of this paper is to carry out the analysis, detection and containment of the Split Personality malware entirely on the virtualized system. We perform dynamic binary instrumentation of the sample under test in order to obtain its low level information as well as to intercept all the API calls made by it. We then check to see if the sample is trying to access any information which would help it in determining the VM presence. If a match is found with any of our monitored set of API calls or low level instructions, our tool logs the activity and provides fake values to the sample so as to make it feel that it is running on the native system.

Proposed Algorithm:

Step 1: *Maintain a list of all the hardware as well as registry querying API calls. Also maintain a list of all the VM specific instructions such as SIDT, SLDT, SGDT, STR, IN.*

Step 2: *Perform dynamic binary instrumentation of the sample under test in order to obtain its low level information as well as to intercept all the API calls made by it.*

Step 3: *Check to see if the sample under test makes a call or executes any of the monitored API calls or instructions respectively. If a match is found, set the OUTPUT to "Split Personality Malware Detected". Also, log the activity and provide fake values to the sample so as to make it feel that it is running on a host system.*

V. CONCLUSION

Users throughout the Internet are plagued by malicious attacks on an on-going basis. The task of defending against these attacks is complicated by many factors, including complexity, scale, and the increasing sophistication of malware authors.

Split Personality malware is on a gradual rise and proactive measures are necessary to curb them at the right time before they become uncontrollable. We found lack of academic research in this field. Moreover there does not exist any full-fledged tool that detects as well as tricks this class of malware to make it feel that it is running on a host OS even when it is running on a virtual one.

REFERENCES

[1]Erbschloe, Michael. Trojans, Worms, and Spyware: A Computer Security Professional's Guide to Malicious Code. Butterworth-Heinemann, 2004.

[2]M. Eichin and J. Rochlis. With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988. In *Proceedings of IEEE Security and Privacy*, Oakland, CA, June 1989.

[3]Vishnani K. (2011), "Introduction to Malware".http://securityresearch.in/index.php/2011/01/projects/malware_lab/introduction-to-Malware/?ubiquitous_id=17 (Jan 20, 2011)

[4]Effcient Detection of Split Personalities in Malware". *In the Proceedings of 17th Annual Network and Distributed System Security Symposium (NDSS 2010),San Diego,* February 2010

[5]OECD, "Malicious Software (Malware): A security threat to Indian economy",(2007) http://www.oecd.org/dataoecd/53/34/40724457.pdf (Oct 20, 2010)

[6]Vishnani K., Pais A., Mohandas R. (Aug, 2011), "Detecting & Defeating Split Personality Malware". In the Proceeding of Fifth International Conference on Emerging Security Information, Systems and Technologies. France, August 2011.

[7]P. Ferrie. "Attacks on Virtual Machines". In Proceedings of the Association of Anti-Virus Asia Researcher Conference, 2007.

[8]Liston T. and Skoudis E. (2006). "On the Cutting Edge: Thwarting Virtual Machine Detec-tion" [Online]. Available: http://handlers.sans.org/tliston/ThwartingVMDetection_Liston_Skoudis.pdf (Nov 1, 2010)

[9]Quist D. and Smith V. (2005). "Detecting the Presence of Virtual Machines Using the Local Data Table" [Online] Available: http://www.offensivecomputing.net/files/active/0/vm.pdf (Nov 14, 2010)

[10]Omella A. (2006). "Methods for Virtual Machine Detection" [Online]. Available: http://www.s21sec.com (Nov 24, 2010)

[11]Klein T. (2005). "Scooby Doo - VMware Fingerprint suite" [Online]. Available: http://www.trapkit.de/research/vmm/scoopydoo/index.ht ml (Nov 20, 2010)

[12] Lau B. and Svajcer V. "Measuring virtual machine detection in malware using DSD tracer". In the Proceedings of Virus Bulletin, 2008, pp. 181-195.