

An Efficient Technique to Destroy Split Personality Malware

Nisha Lalwani

Department of Computer Science
and Engineering

Shri Ramdeaobaba College of
Engineering and Management,
Nagpur, India

M.B.Chandak

Department of Computer Science
and Engineering

Shri Ramdeaobaba College of
Engineering and Management,
Nagpur, India

N.V.Thakur

Department of Computer Science
and Engineering

Shri Ramdeaobaba College of
Engineering and Management,
Nagpur, 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. In the process, if the malware destabilizes the guest OS, they simply discard it and load in a fresh image. However to overcome this technique, malware developers have developed a new class of Malware called as Analysis Aware Malware or Split Personality Malware. This class detects the presence of VM and if it finds any, it hides its malicious nature and behaves like any other normal application. In this paper we will provide an approach to counter Split Personality Malware.

General Terms

Malware Detection, Security.

Keywords

Analysis Aware malware, Split Personality malware, VMware, VM detection, guest OS, host OS.

1. 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.

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 on the virtual machines without having their systems affected. 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 behaviours enabling them to build patches against the vulnerabilities that the malware exploit.

However Malware developers have again added an analysis aware functionality to the malwares. These kinds of malwares analyzes the system in which they are running, if they find that they are running on Virtual Machine, they behave like a normal applications hiding their malicious nature and thus escaping detection. When they are being transferred to the host OS, then they behave like a malware destroying the host machine.

In this paper, we discuss our novel approach using which we detect the VM detection attempts and further trick the malware into believing that they are running on a host OS and hence make them exhibit their malicious nature.

This paper is divided into 5 sections. Section 2 discusses techniques used for VM Detection. In section 3, we discuss the Literature Survey done for Split Personality malware. In section 4 we propose our approach to detect and defeat Split Personality malware. In section 5, we conclude.

2. VM DETECTION TECHNIQUES

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

2.1 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.

2.2 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.

2.3 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.

2.4 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 cached 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!

2.5 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.

3. 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].

3.1 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.

3.2 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].

4. 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.

5. 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.

6. 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.

7. 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] Efficient 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 Detection" [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.html> (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.