CERIAS Tech Report 2015-9
Basic Dynamic Processes Analysis of Malware in Hypervisors Type I & II
by Ibrahim Waziri Jr, Sam Liles
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

BASIC DYNAMIC PROCESSES ANALYSIS OF MALWARE IN HYPERVISORS: TYPE I & II

Ibrahim Waziri Jr
PhD Candidate in Information Security
CERIAS – College of Technology
Purdue University
West Lafayette, IN 47907
iwaziri@purdue.edu

Samuel Liles PhD
Associate Professor
College of Technology
Purdue University
West Lafayette, IN 47907
sliles@purdue.edu

ABSTRACT

In this paper, we compare, analyze and study the behavior of a malware processes within both Type 1 & Type 2 virtualized environments. In other to achieve this we to set up two different virtualized environments and thoroughly analyze each malware processes behavior. The goal is to see if there is a difference between the behaviors of malware within the 2 different architectures. At the end we achieve a result and realized there is no significant difference on how malware processes run and behave on either virtualized environment. However our study is limited to basic analysis using basic tools. An advance analysis with more sophisticated tools could prove otherwise.

Keywords: Basic Malware Analysis, Cyber Forensics, Hypervisor Forensics, Cloud Forensics and Digital Forensics

1. INTRODUCTION

The use of virtualized environments is ubiquitous in malware analysis. The ability to isolate and quickly restore the system to a known configuration after an analysis run is two key features of virtualized environments that facilitate Malware analysis.

Today most industries and organizations solemnly depend on the cloud for their daily activities. Information storage and data sharing has increased the need for cloud demand. With the cloud being a virtualized environment, usually consisting of servers and hypervisors. The need to better protect these devices arises. When we talk about virtualization, the first thing that comes into mind is a virtual machine running on top of an operating system. However we have different forms and types of virtualization environment. This paper presents and discusses about the Type 1 virtualization environment (aka bare-metal hypervisor) with respect to malware analysis. These hypervisors are installed on top of a bare-metal machine. And the virtual machines aka containers are installed on top of these hypervisors.

With the number of malware threats increasing every day, this increases the needs for new malware analysis and study. In 2011 alone, McAfee Labs identified more than six million unique malware samples [1]. Using virtualized environments for malware analysis have been an obvious choice; unfortunately, the malware designers have developed a new class of malware class Virtual Machine aware malware. This VM-aware malware can detect virtualized or emulated environments and change its behavior to suit the analysis.

Majority of analysis are being carried out within a Type 2 hypervisor environment. Now with most cloud consumers and vendors using a Type 1 hypervisor. What happens when a malware is designed and target towards a Type 1 hypervisor? Today, we have fewer people with an in-depth knowledge of how to analyze a malware within a cloud environment. Some detection techniques that can be used to identify I/O ports, specifications, configurations, etc. [2].

For malware analysis to be successful we need to have in-depth knowledge of how the malware behaves. There has been considerable amount of research work focused on the detection and analysis

of malware within a guest OS (These research work will be presented and explained in later chapters). However all previous work have not mentioned or touches the area of malware behavior within a Type 1 environment.

2. LIMITATIONS

This study is limited to comparing basic malware processes analysis on virtualized environments. The study is not a complete of thorough analysis of malware. Other parameters outside the scope of basic processes analysis were not analyzed. Tools and choice of Operating System and devices to be used is explained in the implementation section of this study. The audience that will benefit from the most paper includes majors of the Information Security/Forensic Experts & such as Cyber Forensic experts, Cloud Security Experts, Information Security Experts, Malware Analyst and Students.

3. PREVIOUS STUDY

We look into previous researches that have been made with respect to malware in the cloud. Considering it is still an emerging field, not much has been done with respect to comparison between the different types of hypervisors. However few studies were made which concentrated on analyzing a malware in a virtualized environment.

Some researchers focused on "Computer Forensic Analysis in a Virtual Environment" (Bam, 2007). In this study, they discussed the potential role of virtual environments in the analysis phase of computer forensics investigations. They also identified limitations of virtual environments and made a conclusion that analysis in a virtual environment cannot be considered a replacement for conventional techniques of computer evidence collection and analysis. They took a look at the application of VMware VM in the analysis phase of computer forensics.

Another interesting study is "BareBox: Efficient Malware Analysis on Bare-Metal" (Kirat., Vigna., Kruegel 2011). Considering Bare-Metal is the same as Type I hypervisor. And our study focuses on the analysis of the two different hypervisor. The need to set up both environment arises and this study have guided us on how to setup an analysis environment for Type I hypervisor. In the paper they present the design, implementation and evaluation for malware analysis framework for bare-metal systems that are based on a fast and reboot less system restore technique. They accomplished live system restore by restoring the entire physical memory of the analysis operating system from another small operating that runs outside of the target OS. By using that technique they were able to perform a reboot less restore of a live Windows system running on commodity hardware within four seconds. They also analyzed 42 malware samples from seven different malware families that are known to be silent in a virtualized or emulated environments, and all the malwares showed their true behavior with the type I analysis environment.

Because malware analysis environment is incredibly resource intensive to deploy and it tends to be highly customized requiring extensive configuration to create, control and modify. (Schweiger et, al) did a study "malware analysis on the cloud: increased performance, reliability and flexibility". In this study they attempt to address both the aforementioned concerns by providing an easily deployable, extensible, modifiable and open-source framework to be deployed in a private-cloud based research environment for malware analysis. The framework used was written in Python and is based on the Xen Cloud Platform which utilizes the Xen API allowing for automated deployment of virtual machines, coordination of host machines and overall optimization of resource available. They identified each part of the malware analysis process as a discrete component. Additional functionality and modifications were achieved through the use of custom modules. They also created a sample implementation that includes basic modules for each step of the analysis process including traditional anti-virus checks, dynamic analysis, tool output aggregation and classification. At the end they showed an increase in the performance, reliability and flexibility compared to an equivalent lab environment created without the use of the framework.

"Dynamic Malware Analysis using IntroVirt: a Modified Hypervisor-Based System" (White., et. al 2013) presents a system for dynamic malware analysis which incorporates the use of IntroVirt. An introspective hypervisor architecture and infrastructure that supports advanced analysis techniques for stealth-malware analysis. The

system allows for complete guest monitoring and interaction, including the manipulation and blocking of system calls. IntroVirt is capable of bypassing virtual machine detection capabilities of even the most sophisticated malware by spoofing returns to system call responses.

Others focused on designing "A framework for behavior-based malware analysis in the cloud" (Martignoni et., al). In the study they present a new framework for improving behavior-based analysis of suspicious programs. The framework allows an end-user to delegate security labs, the execution and analysis of a program and to force the program to behave as if it were executed directly in the environment of the former. The evaluation they presented at the end demonstrated that the proposed framework allows security labs to improve the completeness of the analysis, by analyzing the piece of malware on behalf of multiple end-users simultaneously while performing a fine-grained analysis of the behavior of the program.

Based on these previous studies, we decided to follow some aforementioned approaches to implement our analysis environment. Even though these studies focused on malware analysis within virtualized environments. None of the study made an attempt to compare the analysis of a malware within the two different types of hypervisors.

4. SETUP & IMPLEMENTATION

In this section, the type I and type II hypervisor environments set up, the malware analysis, the tools used, the resources and hardware used to achieve the goal of this study are introduced. Before carrying out the dynamic analysis we run a basic static analysis of the malware to give us an idea of what type of malware we are dealing with. We used some basic static malware analysis tools to carryout out the basic static analysis. These tools are:

- Virustotal.com
- PEiD
- Resource Hacker

The tool we used for the basic dynamic analysis is:

- Process Monitor

The hardware and environmental setup tools used are:

- VMware ESXi 5.0 hypervisor
- VMware vSphere Client
- Windows 8.1 64bit Operating System
- VMware Fusion 7 – Hypervisor for OS X.

Type I Environment:

Windows 8.1 operating system with all the tools mentioned used for basic static and dynamic analysis above installed, runs as a virtual machine on a VMware ESXi 5.0 hypervisor installed on a server. A VMware vSphere Client is used to control and monitor the virtual machine. The VMware VSphere is installed inside a Windows 8.1 operating system running on a standalone computer. Figure 1 depicts the Type I Environment infrastructure architecture.

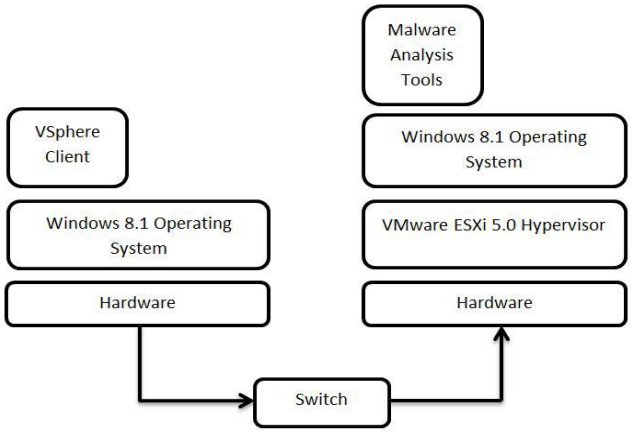

Figure 1: Type I Architecture

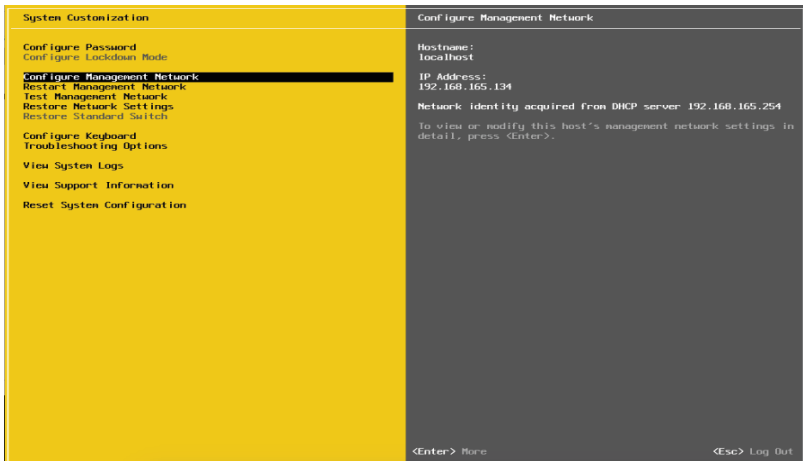

Figure 2: Running ESXi 5.0 Hypervisor

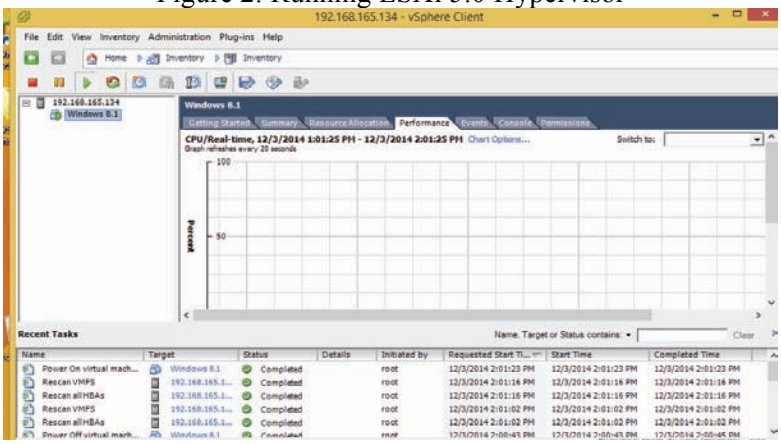

Figure 3: ESXi 5.0 in vSphere Client running Windows 8.1

Type II Environment:

Another Windows 8.1 operating system with all the malware analysis tools runs as a virtual machine within VMware Fusion 7 hypervisor. The VMware Fusion hypervisor is installed on an OS X Yosemite running on a MacBook Pro laptop. Figure 4 shows the Type II environment architecture.

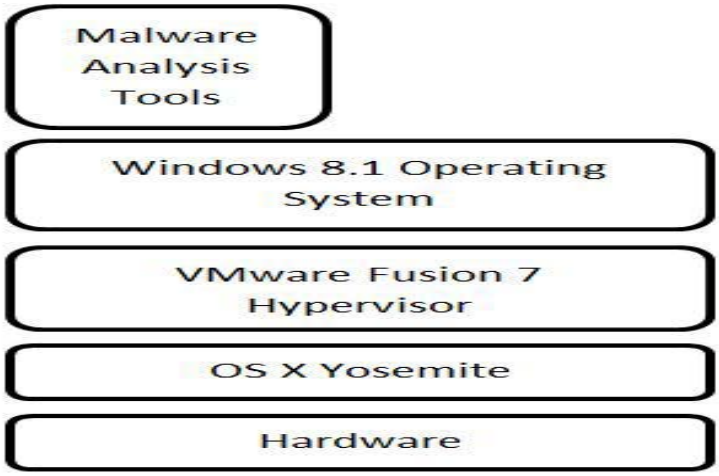

Figure 4: Type II Architecture

Process Analysis Tools:

All the tools used in this study for the malware process analysis are installed on the Windows 8.1 operating system mentioned above.

Malware Source & Extraction:

The malware used in this was downloaded from the free malware repository "MalShare"

5. ANALYSIS & RESULTS

Basic Static Analysis:

Considering we don't have to run the malware to run a basic static analysis, therefore we are certain that the result will be the same on both types of hypervisors. However we still conducted the analysis on both hypervisor platforms. The result appears to be accurate and the same within both environments. In a tabular below, we showed a comparison of the result within both Type I and II environments.

	Virus Total		PEiD		PEview	
	Type I	**Type II**	**Type I**	**Type II**	**Type I**	**Type II**
Virus Signature	✓	✓				
Compilation Date	✓	✓			✓	✓
Import/Exports	✓	✓			✓	✓
File Size	✓	✓			✓	✓
File Type	✓	✓				
Target OS	✓	✓				
Packed/Unpacked			✓	✓		
Compiler			✓	✓		
Section Numbers					✓	✓
Headers					✓	✓
Time Stamp					✓	✓
Dialog					✓	✓
Accelerators					✓	✓
Version Info					✓	✓

Table 1: Basic Static Analysis Comparison

Basic Dynamic Process Analysis:

For this analysis, we have to run the malware before monitoring its processes. We know that the two architecture differs and therefore we run the malware on each environment while as we monitor its processes and behavior. Figures below depict the process running on both environments. The malware appears to behave in the same pattern within the two environments. However there is a little bit of differences between the two. The result of this test is shown in a tabular form below:

	Process Monitor		Process Explorer	
	Type I	**Type II**	**Type I**	**Type II**
Registers			✓	✓
Libraries			✓	✓
Execution Time	✓	✓		
CPU Load	✓	✓		
Average Memory	✓	✓		

Process Name	✓	✓	✓	✓
PID (Process Identifier)			✓	✓
Operation	✓	✓		
Path Address	✓	✓	✓	✓
Category			✓	✓
TID (Thread Identifier)			✓	✓

Table 2: Basic Dynamic Process Comparison

6. CONCLUSION

This study focused on a comparison of malware processes behavior within a Type I and II hypervisor environments. The analysis was meant to see the difference and similarities between the two architectures. Most malware analysis is focused on Type II hypervisors considering it is easier to deploy and maintain. And at any time an analyst can delete a VM if it crashes and run again.

Malwares do not only target individuals, but also companies and organization running huge servers and cloud networks are prone to malware attacks. An attack could be targeted toward the servers running the hypervisors for the cloud. Without an idea of how malware processes run in the cloud. Behavioral analysis might be a difficulty for an analyst. So we therefore decided to carry out this analysis to see if an attack target servers could be analyzed in a standalone environment.

From the analysis result, we see in a tabular form that there is no significant difference between how malware processes run in a Type I environment and that of a Type II. However the analysis we did is just a ground work for malware analysis. To conclude if malware behavior is different or the same within the two types of hypervisors, an advance malware analysis must be carried. This could be a future with respect to this study.

7. REFERENCES

Anderson, B., Quist, D., Neil, J., Storlie, C., & Lane, T. (2011). Graph-based malware detection using dynamic analysis. *J Comput Virol,* 247-247.

Bayer, U., Kirda, E., & Kruegel, C. (2010). Improving the Efficiency of Dynamic Malware Analysis. *SAC,* 1871-1878.

Beaucamps, P., Gnaedig, I., & Marion, J. (2010). Behavior Abstraction in Malware Analysis.*LNCS,* 168-182.

Bem, D., & Huebner, E. (2007). Computer Forensic Analysis in a Virtual Environment. *International Journal of Digital Evidence, 6*(2).

Chen, X., Anderson, J., Bailey, M., & Nazario, J. (2008). Towards an Understanding of Anti-virtualization and Anti-debugging Behavior in Modern Malware. *International Conference on Dependable Systems &Networks*.

Egele, M., Scholte, T., Kirda, E., & Kruegel, C. (2012). A Survey on Automated Dynamic Malware-Analysis Techniques and Tools. *ACM Computing Surveys, 44*(2).

Firdausi, I., Lim, C., Erwin, A., & Nughroho, A. (2010). Analysis of Machine Learning Techniques Used In Behavior-Based Malware Detection. *Second International Conference on Advances in Computing, Control, and Telecommunication Technologies,* 201-203.

King, S., Chen, P., Wang, Y., Verbowski, C., Wang, H., & Lorch, J. (2006). SubVirt: Implementing malware with virtual machines. *IEEE Symposium on Security and Privacy*.

Kirat, D., Vigna, G., & Kruegel, C. (2011). BareBox: Efficient Malware Analysis on Bare-Metal.*ACSAC*.

Kolbitsch, C., Comparetti, P., Kruegel, C., Kirda, E., Zhou, X., & Wang, X. (n.d.). Effective and Efficient Malware Detection at the End Host. *18th USENIX Security Symposium,* 351-366.

Martignoni, L., Paleari, R., & Bruschi, D. (2009). A Framework for Behavior-Based Malware Analysis in the Cloud. *ICISS,* 178-192.

Nguyen, A., Schear, N., Jung, H., Godiyal, A., King, S., & Nguyen, H. (2009). MAVMM: Lightweight and Purpose Built VMM for Malware Analysis. *Annual Computer Security Applications Conference*.

Quist, D., & Liebrock, L. (2009). Visualizing Compiled Executables for Malware Analysis. *6th International Workshop on Visualization for Cyber Security*.

Saydjarl, S. (2009). Hiding Virtualization from Attackers and Malware. *IEEE Security & Privacy,* 63-63.

Schweiger, M., Chung, S., & Endicott-Popovsky, B. (n.d.). Malware Analysis on the Cloud: Increased Performance, Reliability, and Flexibility. *Graduate Capstone - Center for Information Assurance and Cybersecurity, University of Washington*.

Sun, M., Lin, M., Chang, M., Laih, C., & Lin, H. (2011). Malware Virtualization-Resistant Behavior Detection. *IEEE 17th International Conference on Parallel and Distributed Systems*.

Uppal, D., Mehra, V., & Verma, V. (2014). Basic survey on Malware Analysis, Tools and Techniques. *International Journal on Computational Sciences & Applications (IJCSA), 4*(1), 103-112.

White, J., Pape, S., Meily, A., & Gloo, R. (2014). Dynamic Malware Analysis Using IntroVirt: A Modified Hypervisor-Based System. *Spiedigitallibrary.com, 8757*.