A Study on Detection of Hacking and Malware Codes in Bare Metal Hypervisor for Virtualized Internal Environment of Cloud Service

Jung-oh Park

Dept. of Information Communications, DONGYANG MIRAE University, South Korea

Abstract
With rapid rise of virtualization technology from diverse types of cloud computing service, security problems such as data safety and reliability are the issues at stake. Since damage in virtualization layer of cloud service can cause damage on all host (user) tasks, Hypervisor that provides an environment for multiple virtual operating systems can become a target of attack by hackers. This paper proposes a method of detecting existing hacking and malware codes on virtualization technology called Hypervisor (bare metal).

Keywords:
Hypervisor, Virtualization, Bare Metal, Cloud Service

1. INTRODUCTION

Gartner, Inc. prospected that 60% of virtualization servers in the world will have more vulnerable security compared to physical servers until 2012. According to a cloud report on 5,300 subjects from 38 nations, most of corporations were found to place foremost importance on security problem when introducing cloud system [1].
Since cloud service is characterized by common use of resources, information service based on the internet, outsourcing of information system, and diverse device environments, security problem is an obstacle in activation of cloud computing.
Additional security risks are posed with recent development of new technologies such as hardware-based VMM (Hypervisor) AP for hardware virtualization (Intel VT, AMD-V, and etc.) inside virtualization space. Bare metal Hypervisor (full virtualization technique) has advantages of supporting hardware virtualization and I/O efficiency, expected to become a core virtualization technology in cloud services [2][3].
Since security requirements in virtualization during construction of cloud service can differ according to the scope of service, method of provision, and method of embodiment, it is necessary to take measures on additional security risks. Technical security requirements can be divided into outer domain (access control on cloud service users) and inner domain (virtualization domain). While outer domain provides many security solutions based on existing security technologies such as user identification, authentication, and authentication policy, inner domain has a problem of difficulty in applying existing security technologies due to infrastructure of virtualized cloud service.
In order to resolve such problems, Hypervisor technology was analyzed and proposed in this paper as an inner domain of cloud service technology.
Chapter 2 analyzes security vulnerabilities of Hypervisor technology in virtualization. Chapter 3 proposes a virtualization security structure based on bare metal Hypervisor. Chapter 4 is performance analysis and Chapter 5 is conclusion.

2. RELATED WORK

2.1 Analysis on Vulnerabilities of Cloud Service Virtualization Technology

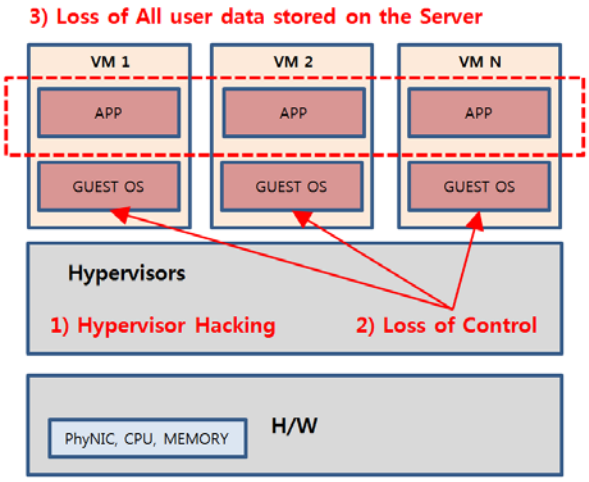

Figure 1. Hacking threat in Hypervisor

NIST (National Institute of Standards and Technology) and CSA (Cloud Security Alliance) analyzed that attackers can use cloud resources immorally or abuse them, causing risks in cloud computing [4][5]. In virtualization

Manuscript received October 5, 2013
Manuscript revised October 20, 2013

environment, existing vulnerabilities such as hacking like denial of service and risks of malware codes can be applied [6][7].
In general, there are vulnerabilities in virtualization: First, loss of control due to hacking of Hypervisor and second, malware code infection of guest OS(Operaing System) from infection of host OS and spread of Hypervisor infection to guest OS [8]. As in Figure 1 and Figure 2, cloud service is characterized by virtualization domain that integrates, redistributes and shares resources. There are problems such as exposure or loss of all data stored in servers and inheritance of system vulnerabilities.
In case of host based Hypervisor shown in Figure 2, all host OS can be infected if malware code or virus communicates through virtual kernel [8].

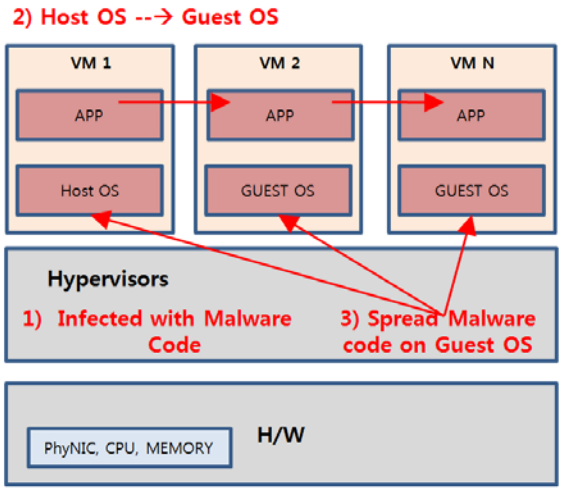

Figure 2. Malware code threat in Hypervisor

When a vaccine is installed in a virtualization OS, the OS becomes defenseless against malware code and virus in the kernel. This is because of difference in method of approach to data inside virtual machine between vaccine application system call in existing original system and system call in virtualization system. Security technologies must be applied at the Hypervisor level, where virtual OS is controlled and actual data communication path is controlled.

2.2 Analysis on Problems of Bare Metal Hypervisor

Host based Hypervisor uses VMM (virtual machine monitor) technology that controls data flow between user mode and kernel mode, as well as privilege rights. Figure 3 and Figure 4 represent host based Hypervisor and bare metal based Hypervisor.
Since VMM is installed on top of OS and all I/O requests in virtualization space are delivered through the host OS, the domain can be used to detect intrusion of malware codes and viruses. In fact, the VMM supports integrated control of virtual OS and can be used as a domain for security management.

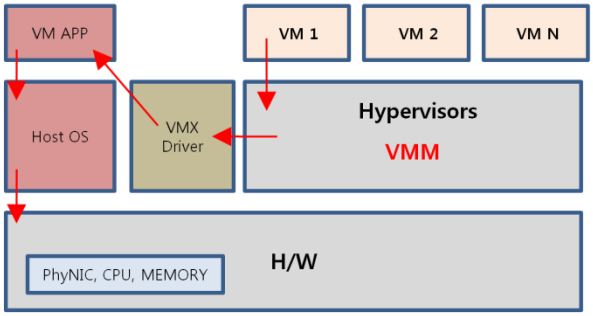

Figure 3. Host based Hypervisor

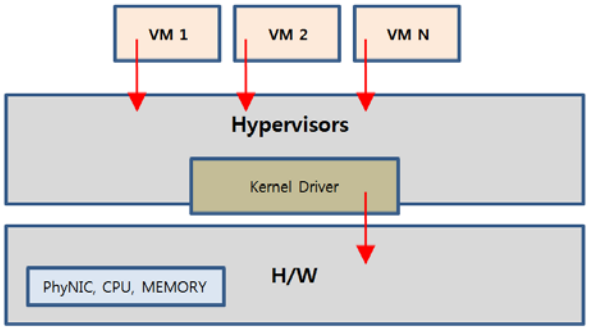

Figure 4. Bare metal based Hypervisor

Bare metal method conducts direct communication with the kernel driver for all I/O requests. There is no domain for controlling data as in host based Hypervisor. Though there are various advantages from efficiency in direct communication with hardware, it is vulnerable in terms of security perspective. Monitoring technology such as VMM level provided by host based Hypervisor is demanded.
Existing commercial products that use VMM technology are subordinate to specific Hypervisor technologies and support specific OS. In this paper, VMMA that can control data at the level of Hypervisor is proposed in order to resolve such compatibility problem of application only in specific systems.

3. VMMA (Virtual Machine Monitor Agent)

In this paper, VMMA (Virtual Machine Monitor Agent) is defined as a bare metal Hypervisor technology for detecting malware codes and viruses. Existing methods of detecting and blocking malware codes and viruses are signature analysis and dynamic analysis. VMMA structure that allows monitoring at the same level as existing VMM

resolves problems of access failure in virtualization (existing detection technique cannot be used) and compatibility. Figure 5 shows the structure of VMMA.

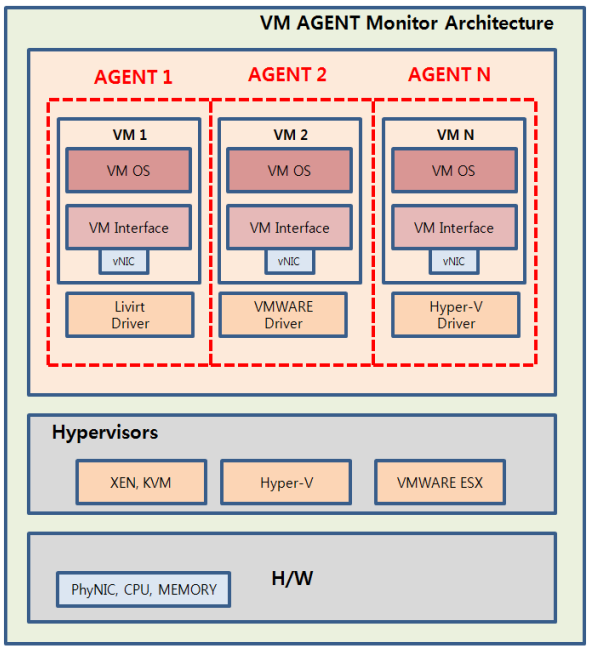

Figure 5. VMMA layer structure

By placing existing hardware kernel driver on top of the agent, performance reduction by communication through OS used by host based Hypervisor as in VMM was minimized. The agent is installed on top of dispersed VM OS and performs operation at the same layer as Hypervisor instead of VM OS interior. The agent supports multiple kernel drivers instead of single kernel driver. Cloud service provider can perform monitoring and integrated control of diverse Hypervisors through integrated management of multiple virtual OS in operation. The agent was designed with a structure that includes separated device driver and OS. Since the agent only performs the role of monitoring TCP/IP packets in data communication, exceptional circumstances are minimized. Figure 6 shows data flow in VMMA in which existing virus and malware code detection technique is applied.

Structure of protected VMMA allows each cloud service provider operating the virtual OS to protect independent / shared data. Each of dispersed agents collects data flow in Hypervisors and reports it to the detection system.

All applications in virtualization environment pass through TCP/IP stack. Traffics occurring from each application approach Hypervisors through the hacking and malware code detection system linked with the agent. Since the detection system is operated independently from Hypervisors, there is no need to additionally embody privilege rights on specific domain. Vulnerabilities to Hypervisor infection are resolved through management of data infected with malware codes and viruses passing thorough existing kernel driver using VM agent.

Figure 7 shows structure of agent in which hacking and malware detection system is applied.

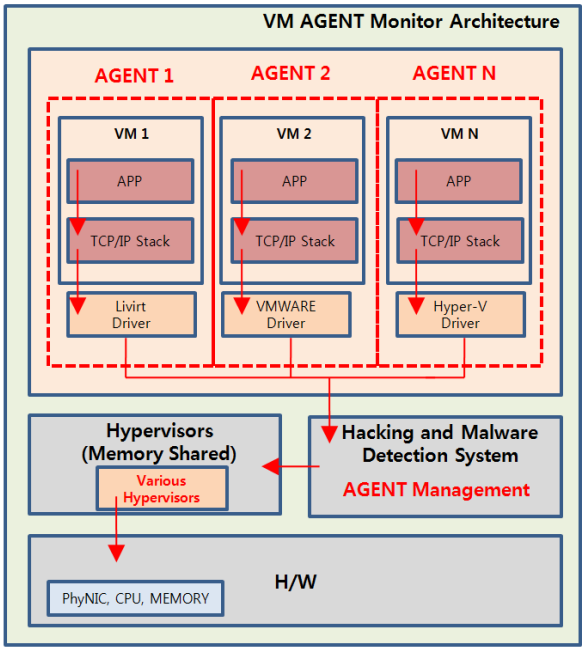

Figure 6. Data flow in protected VMMA

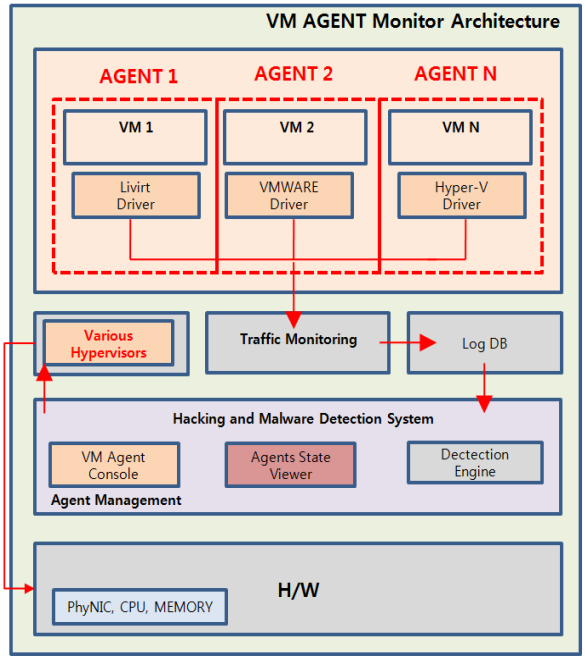

Figure 7. Structure of VMMA with detection system

Logs monitored and collected by each agent can be saved and used in log database located in an actual physical host. The detection system does not have an independent management function due to characteristics of bare metal Hypervisor. It requires a separate management console like VMM console. For the detection engine linked with detection system, existing malware code and hacking detection technologies such as signature analysis and dynamic analysis can be applied.

4. PERFORMANCE ANALYSIS

VMMA proposed in this paper was designed to control and manage safe data flow. Since detection rate of malware codes and hacking is evaluated based on accuracy of detection engine patterns and amount of patterns, performance analysis in this paper was done by comparing performance of kernel call. This is because addition of detection system in intermediate layer can influence performance of virtualization environment. Table 1 shows test environment parameters.

Table 1. Test environment parameters

No	Parameter	Description
1	Cloud OS : Ubuntu 11.10 64bit Server	Openstack Multi Server(2 Servers)
2	VM OS(Total 4 VM Clients)	Windows7, Ubuntu10.04 (2CPU, 1024 memory, 10GB HDD)
3	Hypervisors	Windows(Hyper-V), Ubuntu(KVM)
4	Evaluation List	System Call frequency
		System bottleneck (CPU, Network, Memory)
5	Time	5 Minute

After constructing two servers of openstack based cloud environment, Windows and Linux were generated as VM OS. Items of analysis are frequency of system calls and performance of cloud operation environment. Table 2 and Table 3 are results of system call performance analysis for each VM OS.

For each VM OS, network traffic in port 80 was collected for about five minutes to analyze the number of system calls. In case of Windows 7, the number of system calls was lower by 98 in modified Hypervisor.

Ubuntu showed an increase of 28 calls. This means that the proposed VMMA does not affect system calls. Difference in the number of system calls is caused by irregular traffic generation by traffic generator. VMMA structure was designed to not influence system calls.

Figure 8 and Figure 9 are results of system performance analysis on cloud operation environment.

Table 2. Results of Windows 7 system call performance analysis

Parameters	Original	Modified
Write	6985	6951
Read	186	188
Open	210	209
Close	207	201
Execve	2	2
Access	39	40
Brk	6	6
Munmap	116	105
Mprotect	48	47
Mmap2	206	198
Stat64	114	100
Fstate64	153	127
Set-thread_area	2	2
Total	8274	8176

Table 3. Results of Ubuntu 10.04 system call performance analysis

Parameters	Original	Modified
Write	6897	6921
Read	183	188
Open	201	198
Close	199	191
Execve	2	2
Access	32	34
Brk	5	5
Munmap	102	105
Mprotect	41	40
Mmap2	195	189
State64	102	109
Fstate64	136	141
Set-thread_area	2	2
Total	8097	8125

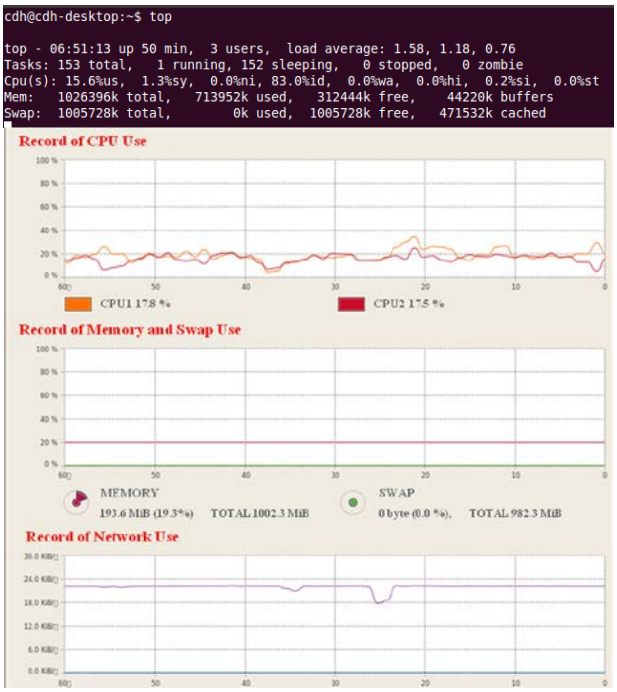

Figure 8. Operation environment of existing cloud service

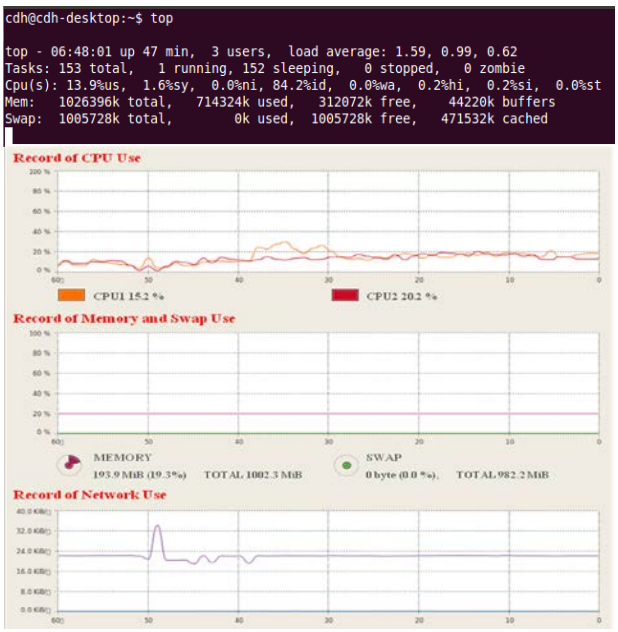

Figure 9. Operation environment of modified cloud service

In this performance analysis, unnecessary daemon processes such as Rwhod and routed were removed for accurate analysis of CPU overload and I/O performance. NICE was used in order to reduce priority of CPU-dependent processes.

As a result of analyzing performance of modified environment, mean CPU usage was under 20% with load average of one minute and five minutes was two or less processes (overload: over 5~10 processes). There was no large difference in CPU load. Mean memory usage was 20%, and swap memory usage was also not influenced.

5. CONCLUSION

In this paper, a VMMA based bare metal Hypervisor was proposed in order to improve vulnerabilities of bare metal Hypervisor method among virtualization technologies of clouds service. The structure was designed to detect and block malware codes and viruses by including kernel driver inside agent. There were advantages of performance efficiency and compatibility of diverse Hypervisor technologies.

As a result of performance analysis, frequency of system calls and performance of operation environment were not influenced by large.

REFERENCES

[1] Orlando Fla, "Gartner Identifies the Top 10 Strategic Technologies for 2012", Gartner, 2011.
[2] Intel, “Intel Virtualization Technology”, http://www.intel.com/technology/itj/2006/v10i3/1-hardware/5-architecture.htm, 2006.
[3] Advanced Micro Devices, “AMD-VTM Nested Paging”, Virtualization(AMD-VTM) Technology White Paper, 2008.
[4] Grance, T., Jansen, W., “Guidelines on Security and Privacy in Public Cloud Computing”, NIST Pubicaion Citation(NIST SP 800-144), 2011.
[5] CSA, “CSA Guidance Version 3”, Security Gudiance for Critical Areas of Focus in Cloud Computing , 2011.
[6] Subashini, S., Kavitha, V., “A survey on Security issues in service delivery models of cloud computing”, ELSEVIER , Journal of Network and Computer Applications , 2010.
[7] Gruschka, N., Jason, M., “Attack surfaces : A Taxonomy for Attacks on Cloud Services”, Cloud Computing(Cloud), IEEE 3rd International Conference on, p 276-279, 2010.
[8] Korea Internet Security Agency, “Cloud Service Information Security Guideline”, Research and Developer Team, 2011.

Jung-Oh Park received the B.S. degree in Computer Science at Sungkyul Univ., Korea, in 2000, and the M.S. degrees in Computer engineering from Myongji Univ., Korea, in 2003, and Ph.D degrees in computer science from Soongsil Univ., Korea, in 2011. His research interests include Network Security, Cryptography and Information Hiding.