A Study on the VMI-based Kernel Runtime Protection System for Virtual Machine in Cloud Environment

Yeo-Reum Jo[1], Ki-Woong Park*
[1]SysCore Lab, Sejong University, Seoul, Korea
sshh802@gmail.com
*Dept of Information Security, and Convergence Engineering for Intelligent Drone
Sejong University, Seoul, Korea
woongbak@sejong.ac.kr

Abstract

Malicious code at the kernel level, such as the kernel rootkit, which manipulates the state of the operating system through code injection and tampering, is a serious threat to system security. These attacks cause a wide range of damage in cloud environments [1]. In cloud infrastructure, multiple virtual machines (VMs) run on a physical machine whose resources are shared through a hypervisor. Attackers consider a VM as a target and use it to attack hypervisors, and other VMs, thereby compromising the entire virtualization infrastructure [2]. Therefore, protecting guest operating systems from such attacks is a major challenge in the cloud environments. Integrity verification can be performed to detect code injection and tampering of the VM kernel area. Several commercial security defense mechanisms are ineffective in a virtualized environment because they generate different hash databases for integrity checks within the individual guest VM and run together, thereby consuming resources. In addition, kernel protection systems must be independent of the VM because it is not safe to protect the kernel with the same privileges as the region in which the kernel malicious code is installed. Therefore, studies have been conducted to protect VMs based on virtual machine introspection (VMI)[3, 4]. VMI is an approach that transparently monitors the execution state of VMs at the hypervisor level. It is not necessary to install an agent within a VM, but it can have access to the memory and CPU registers of the VM. In addition, VMI-based VM protection mechanisms can work in an environment that is completely isolated from the VM, enabling continuous analysis and defense without interference from malicious code. However, these VMI-based virtual machine protection studies are have scalability problems for applications or cause negligible overhead in the memory dump. In this work, we present the overall concept of the system that protect the VM kernel runtime in cloud environments with following requirements - 1) Verify the integrity of VM kernel data to detect tampering: Some data structures in the kernel are used to initialize and configure critical kernel functions, modules, or subsystems. They usually remain unchanged after initialization. Unauthorized modifications to these data structures, especially code pointers in these data structures, will tamper or hijack the original functions or direct to malicious code [5]. Thus, integrity verification should be guaranteed to prevent modification of these kernel data. 2) Guaranteeing integrity validation of the VM kernel image mitigating both pre-injection and post-injection: In case a VM image has already been modified, the detection of kernel tampering through integrity validation is affected. Hence, post-injection must not be permitted using only pre-validated images. 3) Minimizing the overhead of real-time VM introspection: Introspection of a VM causes overhead in reading its physical memory. Because this overhead influences the latency of the guest VM, the overhead of reading the VM memory should be minimized[6]. The structure of the proposed system is shown in Figure 1. It creates a single-reference hash database for a one-kernel-version image for integrity verification. This reduces the time required to verify the integrity of the VMs that use the same kernel version. It also minimizes the verification operation time and reduces the impact of VM performance

The 6th International Symposium on Mobile Internet Security (MobiSec'22), December 15-17, 2022, 2021, Jeju Island, Republic of Korea
Article No. 1, pp. 1-3
*Corresponding author: Dept of Information Security, and Convergence Engineering for Intelligent Drone, Sejong Univ

degradation by dumping data into the kernel protection area of the VM in an in-memory manner. It implements a lightweight VMI that runs at the user level without modifying the host's kernel code, and has minimal functionality for acquiring VM memory and vCPU register information for real-world cloud virtualization infrastructure. In future, we will further improve the performance loss and implement a more efficient system for cloud infrastructure.

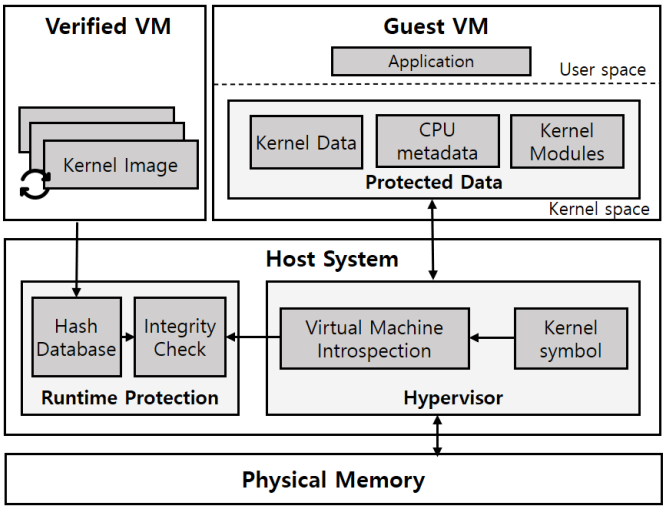

Figure 1: The high-level overview of the proposed system.

Keywords: Cloud Computing, Virtual Machine, Kernel Integrity

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) , South Korea (Project No. RS-2022-00165794, Development of a Multi-Faceted Collection-Analysis-Response Platform for Proactive Response to Ransomware Incidents, 30%, and Project No. 2022-0-007010001003, 30%), and National Research Foundation of Korea (NRF), South Korea grant funded by the Korean government (Project No. NRF-2020R1A2C4002737, 30%), and the ICT R&D Program of MSIT/IITP, South Korea (Project No. 2021-0-01816, Research on Core Technology of Autonomous Twins for Metaverse, South Korea, 10%)

References

[1] MA Ajay Kumara and CD Jaidhar. Virtual machine introspection based spurious process detection in virtualized cloud computing environment. In *2015 international conference on futuristic trends on computational analysis and knowledge management (ABLAZE)*, pages 309–315. IEEE, 2015.

[2] Jakub Szefer, Eric Keller, Ruby B Lee, and Jennifer Rexford. Eliminating the hypervisor attack surface for a more secure cloud. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 401–412, 2011.

[3] Irfan Ahmed, Aleksandar Zoranic, Salman Javaid, and Golden G Richard III. Modchecker: Kernel module integrity checking in the cloud environment. In *2012 41st International Conference on Parallel Processing Workshops*, pages 306–313. IEEE, 2012.

[4] Irfan Ahmed, Aleksandar Zoranic, Salman Javaid, Golden Richard, and Vassil Roussev. Rule-based integrity checking of interrupt descriptor tables in cloud environments. In *IFIP International Conference on Digital Forensics*, pages 305–328. Springer, 2013.

[5] Kunli Lin, Wenqing Liu, Kun Zhang, Haojun Xia, and Bibo Tu. Hyperkrp: A kernel runtime security architecture with a tiny hypervisor on commodity hardware. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2021.

[6] Sang-Hoon Choi and Ki-Woong Park. Cloud-blackbox: Toward practical recording and tracking of vm swarms for multifaceted cloud inspection. *Future Generation Computer Systems*, 137:219–233, 2022.