Volume: 7 Issue: II Month of publication: February

DOI: http://doi.org/10.22214/ijraset.2019.2091

Survey: Virtual Machine Introspection Techniques for Safety Applications for Root kit Attack Recognition

Prof. Anita V. Shinde[1], Ashutosh More[2], Gaurav Nagarkar[3], Pratik Nalawade[4], Rajat Tulshibagwale[5], Sarvesh Rembhotkar[6]

[1]Professor, [2, 3, 4, 5, 6]Student, Dept. of Computer Engineering, Marathwada Mitra Mandal's College of Engineering, Karvenagar, Pune – 411052

Abstract: VM introspection is a approach used to inspect as well as to examine the code that is executed on the particular virtual machine. Virtual machine introspection has gain a special attention in different domain of computer security. Now days, it has been involved in various sectors, including intrusion detection and malware analysis to complete cloud monitoring platforms. This survey involves existing virtual machine introspection tools is required to address different techniques research and focus on some key features that are useful for wide application of virtual machine introspection techniques.

Keywords: Cloud Security, Malware Detection, Rootkit Detection, Virtual Machine Introspection, Virtualization Security.

I. INTRODUCTION

Security and safety are two main features leading future cloud computing research and development. Exploration in virtualization technology has fuelled cloud computing growth and frankly contributed to its improvement. In popular VMI techniques; VM is observing the outcome of introspection is various than the VM being introspected. The main motivation behind VMI is to study each possible variation that happening in a guest OS because of the deployment of a given set of code over its entire lifecycle. Developing and executing VMI-based monitoring applications in public clouds is more challenging compared to self-virtualized data centres. In a self-virtualized data centre, customers have whole control over the private VM. Alas, users in a public cloud computing platform do not have the right to the confidential domain which is in the control of cloud providers. Cloud VMI allows VMI to be offered as a cloud service by a cloud provider to cloud customers. Specifically, Cloud VMI splits the VMI tool development process into two components. The first component that implements the LibVMI functionality to map memory pages is implemented by the cloud provider. The next component that builds the higher level semantics from the lower-level information present on mapped pages is implemented by the customer. In essence, CloudVMI allows cloud users to request introspection privileges for their VMs running on the cloud platform. This design solves the problems:(i) by vitalizing the LibVMI interface and allowing cloud customers to safely invoke it to request introspection obviates the need of having access to the privileged VM. (ii) Need customers to carry their own policies and implementation of monitoring software ease cloud providers from supporting many application and OS distributions and versions?

A. Characteristic Properties of VMI

VMI is applied in widespread domains. Some of the properties are listed below:

1) *Least Performance Impact:* The key objective of virtualization is to contribute resources between existing guests. The implementation of introspection techniques should place as small load as possible on the process of the existing system. Introspection method should not place a load on the hypervisor and real hardware resources.
2) *Minimum Alteration To Hypervisor:* Introspection techniques should work separately and make least modifications to the hypervisor code. This is significant in the application of VMI at the time of minor revisions and in future versions of the VMM.
3) *Intelligibility In The Process:* The action of VMI technique should be transparent to the hypervisor, the guest VM and any program on the guest VM.
4) *Hypervisor Liberty:* The VMI technique should not be relayed on any restricted feature of the hypervisor structural design. It should be valid to any type of hypervisor, irrespective of its execution technology.
5) *No Alteration to Guest OS:* Real-world hypervisors give support to approximately every possible OS as a guest. If the introspection code needs to be changed for each guest OS, its extensive applicability becomes questionable. Even minor revisions and periodical patches to a particular OS may make problems.

B. What is Cloud MI?

The primary goal is to enable cloud operators to provide VMI capabilities as a cloud service, to enable their users to make use of VMI-based tools to monitor their own VMs, without compromising the safety of new cloud users. With Cloud VMI we achieve this goal by virtualizing the low-level LibVMI interface and exposing the virtualized interface as-a-service in a cloud environment.

II. LITERATURE REVIEW

“Mrs. Swarupa Mahesh Deshpande”, et.al [1] introduces The KVM (Kernel-based Virtual Machine Manager) hypervisor manages many VMs (Virtual Machine) on the single host. The virtualization environment for cloud systems in IaaS (Infrastructure as a Service) layer supports heterogeneous operating systems inhomogeneous environment. An Intelligent Virtualization Monitoring System Using KVM monitors the virtual machines under KVM hypervisor. The VMs are the target for the harmful attacks. To protect the VM from the attack, the proposed system uses the VM monitoring script to get the status of the VMs. An intelligent virtualization monitoring system is incorporated as a part of KVM by sending the status of each VM running on the hypervisor.

“Chuliang Weng,” et al. [9] describes in the cloud platform, the startup security of guest virtual machines (VMs) can be guaranteed by existing techniques such as TBoot, and however, how to monitor and guarantee their runtime safety appears to be a non-trivial confront, when they are exposed to the Internet. For a practical cloud system, safety and performance are two significant issues. The dynamic framework called CloudMon to detect kernel rootkits and guarantee the runtime security of guest VMs. Clouded is clear to a guest VM; neither requires its precise system information nor has to one-on-one run with it. In the meantime, Cloud Mon identifies kernel rootkits at the time of self-adjusting monitoring on memory with acceptable overhead.

“Thu Yein Win”, et.al [2] initially focuses on Cyber-attacks that are targeted at virtualization infrastructure. Cloud computing services have turn out to be more and more complicated. This paper introduces a novel approach for malware and rootkit detection system which defends the guests against various attacks. It combines system call monitoring and system call hashing on the guest kernel jointly together with Support Vector Machines (SVM)-based external monitoring on the host.

“Hyun-wookBaek” et al [3] describes virtual machine introspection (VMI) is a mechanism that allows indirect examination and handling of the situation of virtual machines. The indirection of this approach presents an attractive separation property that has resulted in different VMI-based applications mainly includes managing security, performance, and debugging in its environment. As it also require private access to the virtual machine monitor. Unfortunately VMI functionalities are not available for cloud users on the public cloud platforms.

“Yacine Hebbal” et al. [6] Virtual Machine Introspection (VMI) is a technique that enables monitoring of virtual machines at the hypervisor layer. This monitoring concept has increased recently a significant focus in computer security research due to its complete but semantic less visibility on virtual machines activities and isolation from them. VMI works range from addressing the semantic break problem to leveraging explored VMI techniques in order to provide novel hypervisor-based services that belong to different fields.

“Ivan Studnia”, et al [8] discusses the virtualization method which is considered as the heart of Cloud Computing technique, and these methods include their own openness to the traditional as well as connected computer system. It also presents a common idea of these types of vulnerabilities, in extension to manage counter-measures.

III. EXISTING SYSTEM

A. What does Rootkit Mean?

A rootkit is known as venomous type of software that allows an unofficial user to have privileged access to a computer and to restricted areas of its software. A rootkit may contain a number of malicious tools such as key loggers, banking credential stealers, password stealers, antivirus disablers, and bots for DDoS attacks. This software remains hidden in the computer system and let the attacker remotely access to the computer.

B. Types of Rootkits

There are five types of rootkits

1) User-mode rootkits.
2) Kernel-mode rootkits.
3) Boot kits.
4) Hypervisor rootkits.
5) Firmware rootkits.

IV. ADVANTAGES

A. This Approach is useful for Security and safety.

V. CONCLUSION

This survey notices that, in past few years Virtual Machine Introspection technique has grown steadily. VMI- based applications carry out with safety, performance, and debugging in virtual machine environments. Cloud based virtual machine introspection virtualizes its interface and makes it accessible as it is a service in environment of cloud computing. However it grants introspection over the user’s virtual machines that are running in a cloud environment, our VMI is a service abstraction that also allows a new class of cloud-centric.

REFERENCES

[1] Mrs. Swarupa Mahesh Deshpande and Prof. Mrs. BharatiAinapure, “An Intelligent Virtual Machine Monitoring System Using KVM for Reliable And Secure Environment in Cloud”, International Conference on Advances in Electronics, Communication and Computer Technology,(ICAECCT)IEEE, 2016.

[2] Thu Yein Win, HuagloryTianfield, Quentin Mair“Detection of Malware and Kernel-level Rootkits in Cloud Computing Environments”,International Conference on Cyber Security and Cloud Computing IEEE 2015.

[3] Hyun-wookBaek, AbhinavSrivastava and Jacobus Van der Merwe“CloudVMI: Virtual Machine Introspectionas a Cloud Service”, International Conference on Cloud Engineering, IEEE, 2014.

[4] Michael Godfrey, Mohammad Zulkernine“Preventing Cache-Based Side-Channel Attacks in a Cloud Environment,”IEEE Transactions on Cloud Computing - IEEE 2013.

[5] Kara Nance, Brian Hay and Matt Bishop, “Virtual Machine Introspection”, Published by the IEEE Computer Society. IEEE 2015.

[6] Yacine Hebbal ; Sylvie Laniepce ; Jean-Marc Menaud, “Virtual Machine Introspection: Techniques and Applications”, International Conference on Availability, Reliability and Security IEEE, 2015.

[7] Linda Joseph and RajeswariMukesh, “Detection of Malware Attacks on Virtual Machines for a Self-Heal Approach in Cloud Computing using VM Snapshots”, JOURNAL OF COMMUNICATIONS SOFTWARE AND SYSTEMS, VOL. 14, NO. 3, SEPTEMBER 2018

[8] Ivan Studnia, Eric Alata, Yves Deswarte, Mohamed Kaâniche, Vincent Nicomette“Survey of Security Problems in Cloud Computing Virtual Machines”HAL-Cloud and security:threat or opportunity, Nov 2012.

[9] ChuliangWeng, Qian Liu, Kenli Li, and DeqingZou.“CloudMon: Monitoring Virtual Machines in Clouds” IEEE Computer Society.IEEE 2016.

[10] Jonas Pfoh**“**Leveraging Derivative Virtual Machine Introspection Methods for Security Applications”IEEE 2013.

[11] Jonas Pfoh, Christian Schneider, and Claudia Eckert "Nitro Hardware-based System Call Tracing for Virtual Machines" Technics University at M unchen, Munich, Germany – 2011.

[12] Shun-Wen Hsiao, Yeali S. Sun, and Meng Chang Chen“Virtual Machine Introspection Based Malware Behavior Profiling and Family Grouping” “arXiv: 1705.01697v1 [cs.CR] 4 May 2017”.