IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 12, December 2015.
www.ijiset.com
ISSN 2348 – 7968

Securing Virtual Machine via Virtual Machine Introspection

Priyanka Bhongade[#1], Prof. Shridevi C. Karande[#2]
Department of Computer Engineering
Maharashtra Institute of Technology Pune, India

[1]priyanka.bhongade@gmail.com
[2]shridevi.karande@mitpune.edu.in

Abstract—V***irtualization refers to the abstraction between the hardware resources and the software running on a system, making it possible to run multiple operating systems on one physical machine, each one completely separated from the other. It has both technological as well as economical roots, and its adoption is still being driven accordingly, in a race to balance between these two aspects in order to get the most out of it. Enough is being said about the efficiency and the cost savings of this technology, but very little about the security implications it might bring. Many of the security issues in virtualization arise due to the difficulty of inspecting and monitoring a virtual machine continuously, as well as the quality and usefulness of the information that can be monitored and extracted. This monitoring can be achieved by virtual machine introspection (VMI) which is leveraged from hypervisor (VMM-virtual machine monitor). Balancing between security and efficiency cannot be achieved using Network-Based IDS or Host-Based IDS but by VMI***.

Keywords— ***Virtual machine(VM), Virtual machine introspection(VMI), Intrusion detection System(IDS), hypervisor security, cloud security, virtual machine monitor(VMM)***.

I. INTRODUCTION

Cloud computing is an emerging trend in the field of IT providing scalable and flexible services to the end users on demand. Cloud offers services in three levels namely infrastructure, platform and software to meet the needs of different kinds of customers. The key cloud characteristics include multitenancy, location and device independence, elasticity, resource pooling and measured service. The IT companies especially the Small and Medium Scale Businesses are moving onto the cloud which enables them to perform high end computational tasks in a cost effective manner. As more and more IT capabilities can be provided as a service in cloud, security becomes a major concern. Among the numerous attacks that can target the cloud environment.

Installing security software only in the network in the case of an intrusion prevention/detection system, or in the host system that operates and administers the virtual machines within it (Fig 1). This approach decreases the processing power and memory required for the security operations by having a central point for protection enforcement. The network connections between the host and other machines could be easily traced, monitored and decisions based on the central policy could be quickly taken. However, the host machine has minimum visibility inside the virtual machines. Effectively, the internal state, the intercommunication between virtual machines and memory contents cannot be adequately monitored. Installing security software such as antivirus, firewalls, and host intrusion and detection systems in every virtual machine (Fig 2). This method to address the relevant security risks can be regarded as optimal in terms of security, but it is not an efficient solution. Robust protection can be achieved since the security software has a complete view of the internal state of each virtual machine, and its interactions between the host or any other virtual machine. However, sacrifices have to be made on behalf of efficiency because each virtual machine will consume a substantial amount of the processing power and memory. Furthermore, a successful attack on the virtual machine could set all the security software nonfunctional either by disabling or bypassing it, or by installing a rootkit.

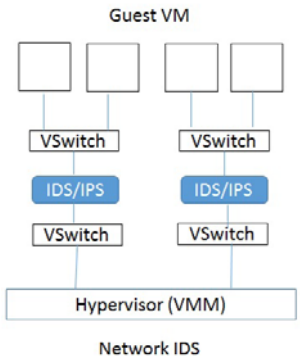

Fig 1 : NIDS

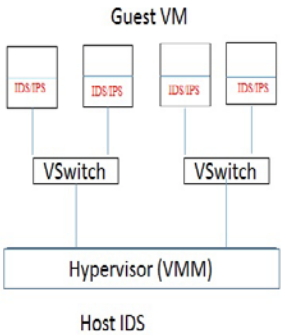

Fig 2 : HIDS

In this paper, we present out of box intrusion detection system that is optimized for virtual system in cloud environment. This paper is organized as follow. First, we provide a background of cloud computing and virtualization in Section 2. In Section 3, we explain virtual machine monitor in detail. Section 4, provides the explanation of need of monitoring in virtualized environment which is followed by section 5 where virtual machine introspection is introduced and section 6 conclude the paper.

ISSN 2348 – 7968

II. VIRTUALIZATION IN CLOUD ENVIRONMENT

Back in time, when the first computer systems were introduced, they were gigantic and expensive systems usually used for critical and demanding operations. The importance of these systems and their critical contribution to day-to-day operations essentially turned them into time-sharing systems in order to be able to take full advantage of their power. The need for multiple users and applications to be run on one physical machine at the same time introduced the idea of system virtualization. Thus, system virtualization is a relatively old concept that incorporates a number of different technologies such as emulation, partitioning, isolation, time-sharing and resource management amongst others, to achieve its objectives.

As a high level description, it is a method used to divide and present the resources of a physical machine (host), as multiple execution virtual machines (guests), by adding a layer of abstraction between the hardware and the applications. The layer of abstraction is usually implemented by software called Virtual Machine Monitor (VMM), that manages the physical hardware resources of the host machine, and make them available to the virtual machines. Essentially, the constraints posed by the system's underlying architecture (IA-32, PowerPC, SPARC) could be circumvented through emulation, thus offering a higher level of flexibility and portability. Virtualization as a step forward from multi-tasking computing to multi-operating system computing. The resources of the guest machines are dependent on the availability of the host machine's resources. Nonetheless, each user of a VM is given the illusion of interacting with a dedicated physical machine as illustrated in Fig 3.

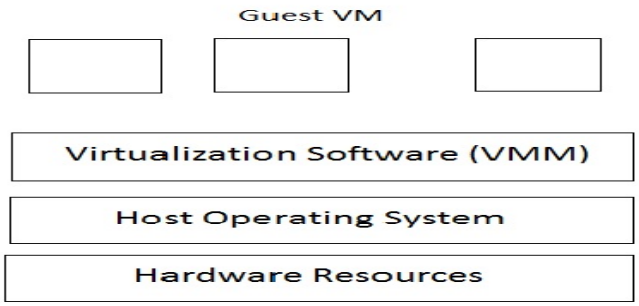

Fig 3 : Virtualization Environment

III. VIRTUAL MACHINE MONITOR

The *Virtual Machine Monitor* (VMM), also known as *hypervisor*, is the core part of every system virtualization solution. Implemented as software or hardware it allows for multiple operating systems to run in a single physical machine. Essentially, the VMM can be seen as a small and light operating system with basic functionality, responsible for controlling the underlying hardware resources and make them available to each guest VM. The more complex an operating system is, the more likely it is to contain bugs or design errors. Therefore, hypervisors should be as minimal and light as possible in order to achieve efficiency and optimal security. All the resources are provided uniformly to each VM, making it possible for VMs to run on any kind of system regardless of its architecture or different subsystems. VMMs have two main tasks to accomplish: enforce isolation between the VMs; manage the resources of the underlying hardware pool. Two different types of hypervisors exist today:

Type I hypervisors also known as *native* or *bare-metal*. Type I hypervisors can be categorized further depending on their design, which can be **monolithic** or **microkernel**.

• **Type II hypervisors** also referred to as *hosted*.

Type I hypervisors run on top of the hardware. Hypervisors of this type are bootable operating systems and depending on their design, may incorporate the device drivers for the communication of the underlying hardware. Type I hypervisors offer optimal efficiency and usually preferred for server virtualization. By placing them on top of the bare hardware, they allow for direct communication with it. The security of the whole system is based on the security capabilities of the hypervisor.

Type II hypervisors are installed on top of the host operating system and run as applications (e.g. VMware Workstation). These hypervisors, allow for creating virtual machines to run on the operating system, which in turn provides the device drivers to be used by the VMs. This type of hypervisors are less efficient comparing to Type I hypervisors, since one extra layer of software is added to the system, making the communications between application and hardware more complex. The security of a system of this type is essentially relying completely on the security of the host operating system. Any breach to the host operating system could potentially result in the complete control over the virtualization layer.

IV. NEED OF MONITORING

It is trivial for a user running programs in a VM to determine if he is operating in a virtual environment, and techniques that guarantee to detect the presence of virtualization software do exist. If a system is found to be virtualized, virtualization-aware malware can change its behavior accordingly, whether by directly attacking the VM and its components, or by attacking the virtualization layer itself (VMM/Hypervisor). A lot of research has been done to find and prevent the means by which malware detects virtualization whether it is due to VMM flaws, certain registry entries, OS quirks or CPU indicators. The impact of a successful attack on the VMM would be severe, putting every VM on the system at risk. The virtualization detection methods are based on the fact that virtual and physical implementations by nature have

ISSN 2348 – 7968

significant differences. These differences are not incidental but agreed to be introduced for virtualization to work efficiently: The need for performance as well as practical engineering limitations necessitate divergences between physical and virtual hardware, both in terms of performance and semantics Apart from retaining security assurance, from the administration's and management's point of view, monitoring is a necessity for ensuring that the environment is healthy and that it functions as it is meant to. It is almost impossible to diagnose problems manually in a virtual environment, since the nested nature of virtualization may mean that problems are not as obvious. A malfunction or a minor problem in one VM could pose many risks on the stability of the others, as well as to the overall integrity of the host machine. Without having the ability to monitor, it is obvious that there is no way to have the assurance that the infrastructure works properly. The nature of virtualization causes functional or security problems to manifest and amplify within the infrastructure. Even minor VM issues can propagate and cause ripple effects to the other VMs that reside on the same host. As we shall discuss later, traditional security practices are not always applicable to virtual environments due to the latter's diverse and one-to-many (host-VMs) relationships. It is important that we understand the ramifications and risks in such environments in order to configure and maintain a successful monitoring program to enable proactive and reactive actions.

The gap between managing an infrastructure and actually protecting it from any kind of malicious attack is filled by utilizing intrusion detection and prevention systems. These systems can be focused on monitoring the network or individual systems' behaviors and their interactions with the existing elements that comprise the infrastructure. Intrusion detection/prevention systems (IDS/IPS) integrate sensors that offer granular analysis and security-oriented inspection methods to prevent or detect system attacks. The intrusion protection system lies within the hypervisor in order to protect the traffic that flows to and from the virtual switches. As mentioned in previous sections, the hypervisor is responsible for mediation between a VM and the host for every requested action, and consequently for access to the outer world. By placing the protection in the hypervisor, all traffic that flows to and from the host and the VMs can be captured and analyzed by the protection system.

The main goal is to present the technologies utilized for protection against security threats in virtualized environments. More specifically, it resides in the region of monitoring techniques which involves mainly preventive and detective actions taken to ensure the normal operation of the virtual machines. Various aspects that affect virtual machine monitoring such as practices, trends, hardware, relevant attacks and others will be examined in order to give a wide view of what surrounds virtual machine monitoring.

The security issues that virtualized environments are facing are more complex to be solved than the ones in traditional environments. That is because a system's activity now, should be monitored on two not so distinct tiers but different for sure. Firstly, the activities that take place in the physical host machine, and secondly, the activities in the residing VMs. Enforcing protection on both of these tiers is critical for ensuring the correct operation of a virtualized environment.

V. VIRTUAL MACHINE INTROSPECTION

VMI technology takes advantage of a privileged guest VM which is responsible for managing the remaining unprivileged VMs residing on a system. When utilizing VMI, the privileged guest gets extra responsibilities and apart from managing the unprivileged VMs, is also responsible for observing their internals and operation. VMI presents a powerful way of determining the specific internal aspects of a guest's execution in a virtualized environment from an external central point.

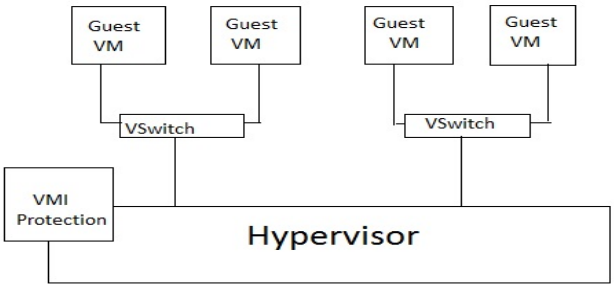

Fig 4 : VMI Protection

The mechanism responsible for facilitating the VMI technology is the VMM component, found in every virtualization system implementation. The VMM is also responsible for creating the layer of abstraction for virtualizing a physical machine's hardware and partitioning it into logically separate VMs. The security assurance offered by the VMM is fundamental for the correct operation of introspection applications. Thus the simplicity and the verification of a VMM is critical to support reliable protection. VMI leverages and benefits from VMM's capabilities in three ways:

1. Isolation. Isolation ensures that, even if the monitored guest is compromised, further access to the VMI application residing in the privileged guest will be prevented. Thus, it should not be possible to tamper with the operation of the IPS.
2. Inspection. The VMM has full access to the residing guests, including memory, CPU registers and I/O operations in order to control them. The deep visibility that the VMM enjoys allows for the complete inspection of a guest and makes it hard for malicious code to evade detection.
3. Interposition. The VMM is able to supervise the VM operation and intercept requests, i.e., privileged CPU instructions, since it mediates between the guests and the host. This functionality can be used by the VMI in order to make decisions for intercepted requests based on a security policy,

ISSN 2348 – 7968

regarding unauthorized or illegal modifications/actions.
isolation is provided by default due to the VMMs functionality. Inspection and interposition require minor modifications to a VMM's code and consideration of possible trade-offs due to integration. Other fundamental VMM capabilities also contribute to the effectiveness of VMI-based applications. For example, the fact that the whole VM state is saved in the VM's files, VMI tools can use checkpoints within these files to be analyzed, make comparisons between the state of compromised and clean VMs, take snapshots for later analysis and so on. The security features that can be offered by the VMI technology have been well understood by the industry, and major players in virtualization have implemented introspection APIs to support their products. In order to detect malicious acts, and more importantly to prevent them for materializing, all VMI-based applications need to somehow gather information about the monitored guest. Further distinctions can be made as to the way a VMI-based solution gathers data, in order to construct the level of semantic awareness it needs, pertaining to the guest OS. When gathering information, techniques like kernel dumping cannot be considered effective in many cases since they perturb the guest's OS execution. Furthermore, considering that dump tools normally belong to an OS which can be potentially compromised, the acquired information cannot be considered reliable. In this section we focus on less intrusive techniques and discuss how are they used for gathering information from a guest VM. These techniques follow the logic of observe-and reconstruct. Inevitably, every VMI-based application will, at some point, need to get information on-the-fly by examining the VM's volatile memory. The reconstruction of the semantics is achieved by putting together such information.

VI. CONCLUSION

The key challenge with VMI is the semantic gap between an external (hypervisor view) and an internal observation (VM view). Another limitation is the performance impact, which is tightly dependent on the deployed environment. VMI's semantic gap refers to the available information two entities gain while observing the same VM. More specifically, an observer observing a VM from the outside (introspection) can see memory pages, registers and generic low level events. An inside observer on the other hand, is able to see semantic level elements such as processes or files and specific events such as system calls. Since VMI-based applications operate in a different context to the guest OS they protect (i.e. outside the guest), they cannot rely on getting the semantic information directly from it. That is because information gained this way is likely to be unreliable if the guest OS is compromised. To that end, VMI tools need to parse the low-level data they are able to see from the guest OS and reconstruct the semantic information themselves

ACKNOWLEDGMENT

We express our sincere thanks to all the authors, whose papers in the area of Virtual Machine Monitoring are published in various conference proceedings and journals, and to all authors and organizations of referred websites.

REFERENCES

[1] Sheng-Wei Lee , Fang Yu "Securing KVM-Based Cloud Systems via Virtualization Introspection" , 2014 47th Hawaii International Conference on System Sciences (HICSS),IEEE Jan. 2014 Page(s):5028 – 5037 .

[2] Gehana Booth , Andrew Soknacki , Anil Somayaji "Cloud Security: Attacks and Current Defenses " 8 th ANNUAL SYMPOSIUM ON INFORMATION ASSURANCE (ASIA'13), June 2013, ALBANY, NY

[3] Tongwook Hwang,Kyungho Son,Youngsang Shin,Haeryong Park "Design of a Hypervisor-based Rootkit Detection Method for Virtualized Systems in Cloud Computing Environments " , Dec 2013 AASRI Winter International Conference on Engineering and Technology (AASRI-WIET 2013)

[4] Nils Gruschka ,Meiko Jensen "Attack Surfaces: A Taxonomy for Attacks on Cloud Services" , July 2010 3rd International Conference on Cloud Computing (CLOUD), IEEE , Pages : 276 – 279

[5] Chimere Barron, Huiming Yu , Justin Zhan " Cloud Computing Security Case Studies and Research " Proceedings of the World Congress on Engineering 2013 Vol II, WCE 2013, July 2013, London, U.K.

[6] Dr.V.Venkatesa Kumar , M.Nithya "Improving Security Issues and Security Attacks in Cloud Computing" , IJARCCE , Vol.3, issue 10 , Oct 2014.

[7] Nagaraju kilari , Dr.R.Sridaran "An Overview of DDoS Attacks in cloud Environment " , International Journal of Advanced Networking Applications(IJANA)

[8] " Meeting the Challenges of Virtualization Security " Trend Micro Aug 2009.

[9] Available: https://en.wikipedia.org/wiki/VMI