

Identifying Trusted Virtual Machines

Extended Abstract for Xen Summit, June 2008

F. John Krautheim, Dhananjay S. Phatak, Alan T. Sherman

University of Maryland, Baltimore County (UMBC)

Baltimore, MD 21250

{kraut1, phatak, sherman} @umbc.edu

Introduction

Virtual machines (VMs) provide many advantages to increase efficiency and decrease cost to data centers; however, these VMs have significant security implications. When software is operating within a virtual environment, the question of which VM is operating arises and whether the configuration of the VM has changed. Additionally, vulnerabilities specific to virtual machines (e.g. *virtual machine based rootkits* (VMBRs) [1]) threaten to subvert the built-in defenses of the guest operating systems.

Virtual machine technology allows a computing platform to create a virtualized system level environment so that an operating system or other program written to run on a real machine believes it running on the bare machine by itself. A *virtual machine monitor* (VMM), or *hypervisor* (HV), is a software solution used to implement the virtualization and control the resources allocated to the guest operating system within the VM. As computing technology becomes increasingly more compact and powerful, the ability for even small desktop machines and portable devices to run multiple virtual environments is increasing. It is foreseeable that virtualized environments could become commonplace or even standard configurations for many computing system.

A *trusted virtual machine* (TVM) is a VM that performs tasks without subverting the security policies of the machine owner, the software licenses, content DRM policies, and user privacy. It also does not harm or eavesdrop on other VMs, the host environment, or hypervisor. In order to trust the virtual execution environment, a method is needed to measure the environment to ensure that the VM meets the all of the TVM requirements and report that measurement to a *policy decision point* (PDP) to authorize a VM for operation. The entire process of measuring a configuration and reliably reporting the measurement is *attestation* [2].

We examine the need for a method to attest that a virtual machine is trusted. These methods rely on the hardware protection mechanisms such as *trusted platform modules* (TPMs), secure virtual machine technology, secure execution technology, and other devices. Leveraging these devices also presents other

challenges to privacy and monitoring that we need to consider as well.

Trusting Virtual Machines

As more data centers move to virtualization architectures, the security and integrity of the virtual machines becomes an increasing concern [3]. We wish to place a level of trust in the VM that the machine is not malicious and will not compromise the integrity of the data and applications contained within the VM.

Several questions arise that place doubt on the trust level of the VM. What assurances are given that the operating environment these virtual machines provide proper protection for guest operating system and will not harm or subvert the guest OS? What happens to the integrity and privacy of the data within the VMs? What assurances are given that a system operating within one of the virtual environments will not harm the environment, other guest OSes, or the VMM? These are the questions of trust in the VM that do not currently have satisfactory answers.

The Need for Identifying Trusted Virtual Machines

There are many situations where a data owner would like to know whether a VM is a member of a group. For example, a Virtual Community of Interest Network (vCOIN) is a network of virtual machines stood up for a specific task or event. In this situation, a master VM images would be created by the data owner, preconfigured with the necessary software and encryption keys for VPNs and data stores, and then distributed to the community of interest members. This allows for rapid deployment and membership in the COI for multi-agency task forces, special event management, and any other situation where multiple dispersed members require a virtual workcenter operate to efficiently and in expeditiously.

In the vCOIN environment, when a VM wishes to connect to the virtual network, it should be identified before authorization is given to access the network. If all VMs are merely copies of the original master VM, then it would be impossible to determine a legitimate

VM from an illegitimate copy obtained by an attacker. Once the attacker joins the network, he would have the same access privilege as the authorized users. To prevent this situation, we would like to authorize the VM through an identification and authentication process. The process should provide information about the VM, such as where it is operating, what software it is running, and who is operating it. Without some form of attestation, it would be impossible to determine this information.

Trusted Hardware's Application to VMs

Trusted hardware is one technology we can leverage to achieve our goals of building trusted virtual machines. This technology provides mechanisms to measure and securely execute software through hardware enforcement. We consider applications for using this technology to help build better trust models for virtual execution environments.

The *Trusted Computing Group* (TCG) has released specifications for trusted hardware to help increase the security of computing systems. The TCG specification for x86 based trusted platforms places the root of trust in a *trusted platform module* (TPM) and uses hardware enforcement mechanisms through extensions to the x86 architecture to protect the execution environment from operating unauthorized code [4]. These protection mechanisms ensure that the correct HV is loaded at boot by measuring the VMM via an authenticated code module [5]. We use Intel's vPro implementation of the TCG specification to enforce security through hardware and to bind VM identities to each VM instance.

Secure Virtual Machine Technology

There are multiple research efforts in building software solutions to secure virtual machines. We can leverage several of these approaches for our goals. A hypervisor can be used to secure applications within the virtual machines [6, 7]. However, we still need to augment these approaches since they do not secure the hypervisor or host operating system against attacks which can compromise the system's integrity. We intend to use components of these research efforts to achieve our end goal of trusted virtual execution environments.

A major component of our effort will employ IBM's sHype operating system independent hypervisor security architecture. sHype secures communications between virtual machines running on the same hypervisor system [8]. We plan to leverage sHype to isolate instances of VMs operating on the platform from each other. We hope to improve the security mechanisms of sHype to protect the hypervisor and guest virtual machines from attack in

the form of altering state or protection against state manipulations.

Trusted Virtual Machines Identification

We propose the technique of *Trusted Virtual Machine Identification* (TVMI) as a method of managing and detecting VMs using *virtual TPMs* (vTPMs) on trusted hardware. We believe TVMI will provide a more robust method for trusted virtual machine management. This research is currently being performed at UMBC's Center for Information Security and Assurance.

There are two different approaches to attestation we can leverage for our purposes: direct anonymous attestation [9] and property based attestation [10]. We utilize vTPMs to create unique signatures or *identities* for each instance of a TVM. The TVMI approach we propose differs from the previous attestation schemes by providing a unique identity for a VM instance that stays with it throughout its lifetime (creation through deletion, including migration). This identity is bound to the properties of the environment to form an identity profile. In the event of duplication, a child identity could be created that is related to the parent, much like DNA. The identity profile should change if security properties of the platform are modified, the machine is migrated, and for each new instance of the VM giving each instance a unique "life experience" that can further disambiguate the attestation to the PDP.

Some of the insight we wish to gain from this effort is to determine the difficulty in uniquely identifying each VM instance and detecting duplicated and migrated VMs in a virtual community of interest network. TVMI is the first phase of a larger project called Virtual Centurion [11]. The ultimate goal of the Virtual Centurion Project is to provide a fully trusted environment for managing and operating VMs for use in vCOINs, multi-level security systems, and other applications where high-integrity environments are required for protecting information and privacy.

We believe that through the use of commercially available trusted hardware, we can develop highly robust mechanisms to identify and manage VMs and detect malicious behavior in virtualized platforms. Trusted Virtual Machine Identification is one part of our multi-faceted approach for trusted virtual machine management.

References

- [1] S. T. King, P. M. Chen, Y.-M. Wang, C. Verbowski, H. J. Wang, and J. R. Lorch, "SubVirt: Implementing Malware with Virtual Machines," in *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, Oakland, CA, 2006.
- [2] D. Grawrock, *The Intel Safer Computing Initiative*. Hillsboro, OR: Intel Press, 2006.
- [3] S. Berger, R. Cáceres, D. Pendarakis, R. Sailer, and E. Valdez, "TVDC: Managing Security in the Trusted Virtual Datacenter," *ACM SIGOPS Operating Systems Review*, vol. 42, pp. 40-47, 2008.
- [4] "TCG Specification Architecture Overview," Trusted Computing Group August 2, 2007.
- [5] D. Challener, K. Yoder, R. Catherman, D. Stanford, and L. van Doorn, *A Practical Guide to Trusted Computing*. Boston: IBM Press, 2008.
- [6] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh, "Terra: A Virtual Machine-Based Platform for Trusted Computing," in *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, Bolton Landing, NY, 2003.
- [7] T. Mitchem, R. Lu, and R. O'Brien, "Using Kernel Hypervisors to Secure Applications," in *Proceedings of the 13th Annual Computer Security Applications Conference*, Los Alamitos, CA, 1997, p. 175.
- [8] R. Sailer, E. Valdez, T. Jaeger, R. Perez, L. van Doorn, J. L. Griffin, and S. Berger, "sHype: Secure Hypervisor Approach to Trusted Virtualized Systems," IBM, Yorktown Heights, NY 2005.
- [9] E. Brickell, J. Camenisch, and L. Chen, "Direct Anonymous Attestation," *Cryptology ePrint Archive, Report 2004/20*, 2004; <http://eprint.iacr.org/2004/205>.
- [10] L. Chen, R. Landfermann, H. Löhr, M. Rohe, and A.-R. S. C. Stüble, "A Protocol for Property-Based Attestation," in *Proceedings of The First ACM Workshop on Scalable Trusted Computing* Fairfax, VA: ACM, 2006.
- [11] "The Virtual Centurion Project," <http://cyberlab.cs.umbc.edu/>.