Accepted Manuscript

Peek-a-Boo: Inferring Program Behaviors in a Virtualized Infrastructure without Introspection

Sanghyun Hong, Alina Nicolae, Abhinav Srivastava, Tudor Dumitraş

PII:	S0167-4048(18)30244-X
DOI:	https://doi.org/10.1016/j.cose.2018.08.010
Reference:	COSE 1390

To appear in: *Computers & Security*

Received date:	21 March 2018
Revised date:	13 August 2018
Accepted date:	31 August 2018

Please cite this article as: Sanghyun Hong, Alina Nicolae, Abhinav Srivastava, Tudor Dumitraş, Peek-a-Boo: Inferring Program Behaviors in a Virtualized Infrastructure without Introspection, *Computers & Security* (2018), doi: https://doi.org/10.1016/j.cose.2018.08.010

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Peek-a-Boo: Inferring Program Behaviors in a Virtualized Infrastructure without Introspection

Sanghyun Hong[a,*], Alina Nicolae[b], Abhinav Srivastava[c], Tudor Dumitraș[a]

[a]*Department of Computer Science, University of Maryland College Park, 8223 Paint Branch Dr., College Park, MD 20742, USA.*
[b]*Department of Computer Science, Aalto University, Konemiehentie 2, 02150 Espoo, Finland.*
[c]*Frame.io, 45 West 27th Street, 10th Floor, New York, NY 10001, USA.*

Abstract

Cloud service providers are often prohibited from accessing the content of tenant VMs, yet current techniques for monitoring attacks and unauthorized activities rely on virtual machine introspection (VMI). While the introspections are useful for narrowing down the semantic gap between the status observed at the hypervisor-level and that seen in a VM, they potentially reveal the sensitive information of a tenant stored in the machine. In this paper, we aim to infer specific program activities in a VM without VMI methods, where our approach has to solve the *strong semantic gap* problem. We introduce *Infermatic*, a system that utilizes only hypervisor-level features and supervised machine learning methods to infer program behaviors in a VM. Using the classifiers trained by *Infermatic*, we can also bridge the strong semantic gap by systematically identifying the semantic links between our hypervisor features and selected program behaviors. In evaluations, we demonstrate that the hypervisor features are effective in isolating program activities and do so with an average accuracy of 0.875 (AUC) for the 24 behaviors that we have identified. In addition, our statistical models (or trained classifiers) can identify the hypervisor features that accurately characterize selected program behaviors when they involve lower-level operations. We further extend *Infermatic*'s ability to detect program behaviors to other security applications—we present a malicious VM detector for the cloud that achieves an average detection of 0.817 (AUC). Our detector shows the hypervisor features are resilient against evasion attacks even when an attacker can reduce the number of available features to the system. Moreover, we present that the detector can operate in a scalable manner by identifying a malicious VM even when the VM under inspection is collocated with other VM's operating under workloads.

Keywords: Cloud Security, Strong Semantic Gap, Program Behavior Detection, Machine Learning, VM Introspection

ACCEPTED MANUSCRIPT

1. Introduction

Cloud computing has become an attractive solution for enterprises because it spares them the efforts of system administration and maintenance. The services delivered by the cloud providers are based on a simple and flexible virtual machine (VM) provisioning technique. Unfortunately, these services do not extend to ensuring the security of virtualized infrastructures, as the task of detecting malware infections and data breaches largely remains the responsibility of customers [1]. Moreover, attackers also benefit from the economies of scale, as VMs are often derived from similar machine images, and hence they likely share many vulnerabilities and misconfigurations. It facilitates the propagation of malware within the cloud infrastructure and makes the cloud a feasible target for attackers.

However, it is challenging for cloud providers to implement a seamless surveillance service because of the *semantic gap* [2]. Existing methods for VM monitoring tackle this problem by utilizing *virtual machine introspection (VMI)* [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17], or by relying on *agents inside the guest operating systems* [18] that gather data from the Linux Audit [19] or Windows Event Tracing [20] frameworks. While introspection methods are easy to deploy, as they do not require instrumenting the customer VMs, they are brittle and hard to generalize. These methods require reverse engineering guest OS abstractions and keeping up with the changes occurring within the OS internals. In-VM agents are more robust to OS evolution since they use system auditing APIs rather than employ the internal data structures. However, they are difficult to deploy consistently in all tenant VMs and are susceptible to tampering by malware executing inside the VM.

Moreover, cloud service providers are often prohibited contractually from accessing the content of tenant VMs. Both introspection-based and agent-based monitoring methods access sensitive customer information, such as the processes running inside a VM or content of the memory and disk. Also, the increasing adoption of encryption for network traffic renders the network intrusion detection systems (NIDS) less effective. Those constraints limit the abilities of introspection methods by substantially reducing the available information in a VM. In consequence, cloud providers who try to identify malicious activities inside tenant VMs start to face the *strong semantic gap* [17] between the collectible data and the necessary information for the

*Corresponding author
Email addresses: `shhong@cs.umd.edu` (Sanghyun Hong), `alina-elena.nicolae@aalto.fi` (Alina Nicolae), `abhinav@frame.io` (Abhinav Srivastava), `tdumitras@umiacs.umd.edu` (Tudor Dumitraș)

Download English Version:

https://daneshyari.com/en/article/11031617

Download Persian Version:

https://daneshyari.com/article/11031617

Daneshyari.com