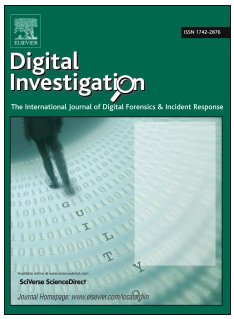
Accepted Manuscript



Leveraging virtual machine introspection with memory forensics to detect and characterize unknown malware using machine learning techniques at hypervisor

M.A. Ajay Kumara, C.D. Jaidhar

PII: S1742-2876(17)30332-8

DOI: 10.1016/j.diin.2017.10.004

Reference: DIIN 714

To appear in: *Digital Investigation*

Received Date: 2 September 2016

Revised Date: 30 September 2017

Accepted Date: 10 October 2017

Please cite this article as: Ajay Kumara MA, Jaidhar CD, Leveraging virtual machine introspection with memory forensics to detect and characterize unknown malware using machine learning techniques at hypervisor, *Digital Investigation* (2017), doi: 10.1016/j.diin.2017.10.004.

Leveraging virtual machine introspection with memory forensics to detect and characterize unknown malware using machine learning techniques at hypervisor

Ajay Kumara M.A. and Jaidhar C.D.

Department of Information Technology, National Institute of Technology Karnataka, Surathkal, India

ajayit13f01@nitk.edu.in, jaidharcd@nitk.edu.in

Abstract

The Virtual Machine Introspection (VMI) has emerged as a fine-grained, out-of-VM security solution that detects malware by introspecting and reconstructing the volatile memory state of the live guest Operating System (OS). Specifically, it functions by the Virtual Machine Monitor (VMM), or hypervisor. The reconstructed semantic details obtained by the VMI are available in a combination of benign and malicious states at the hypervisor. In order to distinguish between these two states, the existing out-of-VM security solutions require extensive manual analysis. In this paper, we propose an advanced VMM-based, guest-assisted Automated Internal-and-External (A-IntExt) introspection system by leveraging VMI, Memory Forensics Analysis (MFA), and machine learning techniques at the hypervisor. Further, we use the VMI-based technique to introspect digital artifacts of the live guest OS to obtain a semantic view of the processes details. We implemented an Intelligent Cross View Analyzer (ICVA) and implanted it into our proposed A-IntExt system, which examines the data supplied by the VMI to detect hidden, dead, and dubious processes, while also predicting early symptoms of malware execution on the introspected guest OS in a timely manner. Machine learning techniques are used to analyze the executables that are mined and extracted using MFA-based techniques and ascertain the malicious executables. The practicality of the A-IntExt system is evaluated by executing large real-world malware and benign executables onto the live guest OSs. The evaluation results achieved 99.55% accuracy and 0.004 False Positive Rate (FPR) on the 10-fold cross-validation to detect unknown malware on the generated dataset. Additionally, the proposed system was validated against other benchmarked malware datasets and the A-IntExt system outperforms the detection of real-world malware at the VMM with performance exceeding 6.3%.

Keywords: Virtual machine monitor, Virtual machine introspection, Memory forensics analysis, Malware detection, Feature selection methods, Machine learning techniques, Semantic gap.

1. Introduction

Due to the propagation of cloud computing, Virtual Machines (VMs) remain attractive targets for cyber crooks, due to easy access from Cloud Service Providers (CSPs) (Pearce et al., 2013). The current generation of malware uses code obfuscation techniques (Lin and Stamp, 2011) and rootkit functionalities (Goudey, 2012) to subvert most of the existing in-host or VM-based anti-malware security solutions to access to the targeted machine. With successful penetration, these malware operate by: (a) leveraging uncovered vulnerabilities of the guest OS to perform illegitimate activities and (b) attacking other VMs running on the same virtualized infrastructure. Preserving the VMs by detecting such sophisticated malware is a challenging task for the CSP (James, 2010).

Traditional in-host or VM-based malware defensive solutions are not virtualization-aware; these solutions often rely on the signature-based technique and are vulnerable to unknown malware that uses zero-day exploits. To address this critical issue, the VMI (Garfinkel and Rosenblum, 2003) has evolved into a promising security solution to introspect untrustworthy guest OSs by operating at the VMM. The useful isolation and inspection properties of the VMM ensure that VMI-based security solutions are secure and tamper-resistant. For example, an `isolation` property leveraged by the VMI confirms that the malicious software running on the introspected VM cannot access or tamper the VMI solution running on the VMM, even though the malware has completely corrupted the guest OS. The inspection property facilitates the VMI to examine the entire run state of the guest OS such as memory, CPUs, registers, I/O, etc.(Zhao et al., 2009). When the VMI introspects a guest OS, it intercepts memory state to reconstruct a guest OS abstraction from the raw memory. However, intercepting low-level details of the current run state of the guest OS and converting it into a meaningful form (e.g., processes, modules, or system calls) presents an obstacle referred to as the *semantic gap* (Dolan-Gavitt et al., 2011a; Jain et al., 2014; Fu and Lin, 2012). Existing VMI-based techniques are not sufficient to reconstruct the high-level rich semantic view of the large kernel data structure (e.g., registry, file system, or kernel object) of the introspected VM that is constantly manipulated by sophisticated malware or rootkits (Prakash et al., 2013). To develop