Visualization-supported Analysis of System Data for Controlled VMI-based Intrusion Detection

Noëlle Rakotondravony, Hans P. Reiser
{nr,hr}@sec.uni-passau.de

Virtual Machine Introspection (VMI) is an approach in which a Virtual Machine (VM) is inspected from outside (from the Virtual Machine Monitor (VMM) or another VM) in order to analyze its running programs. Complete view and information on the state of hosted VMs are available at the VMM level which manages the physical resources allocated to all guest VMs. Inspection purposes can be periodical extraction of information for forensics analysis, continuous monitoring of the execution of the VM and malicious activity detection. In all cases, even if the inspected VM is corrupted, VMI ensures reliable diagnosis since the introspection mechanisms are isolated. One particular way using VMI for monitoring activities of VMs is system call tracing. This method allows to detect a malicious program in a VM, by inserting breakpoints into the running VM and analyzing its extracted system call parameters.

In a general way, VMI mechanisms can produce large amount of low-level data. To exploit their relevance in security analysis, such data are stored and pre-processed in order to provide human-interpretable information through visualization techniques. Existing security visualization mechanisms provide the human user advanced possibilities for exploring different graphs or representation as well as the detailed security context of monitored VMs, to support him in understanding and analyzing the security of his infrastructure.

In such exploration, additional information can become needed for more complete insight into the system state. This requires the user to trigger additional monitoring actions. However, as previously shown, the collection of data from VMI-based mechanisms can induce certain overhead on the performance of the monitored VM. In our work, we aim at accurately evaluating such overhead of the different VMI-based monitoring mechanisms. The research goal of the present work is to model the performance overhead induced by the combination of different introspection actions, thus the impact of any on-demand introspection on the monitored VM. By making the different measurements available, user can wisely control and tailor his analysis while still not alter the proper functioning of the services hosted by the monitored VMs.