Poster: Reconfigurable monitoring and performance awareness in VMI-based SIEM systems

Noëlle Rakotondravony, Benjamin Taubmann, Stewart Sentanoe, Hans P. Reiser
University of Passau, Germany
Email: {nr, bt, se, hr}@sec.uni-passau.de

I. Problem statement

Security information and event management (SIEM) systems are commonly used to enhance system security and handle cyber attacks. Many practical SIEM systems use in-system agents on the monitored system resources for collecting state information that is used for the detection and analysis of incidents. A malicious attacker that gains full control over a system component usually can disable or manipulate such data collection.

Virtual machine introspection (VMI) isolates the monitoring from the monitored entities, enabling data collection that is better protected against malicious manipulations [1]. The DINGfest architecture [2] demonstrates the feasibility of using VMI-based tracing mechanisms in a SIEM architecture. It incorporates VMI-based data acquisition based on the Cloud-Phylactor model [3], a data analysis component that detects suspicious user behaviors and incorporates visual analysis [2], and means for digital forensics and incident reporting.

VMI-based analysis in a SIEM system not only has advantages, but also some shortcomings. In particular, VMI-based mechanisms can potentially be resource intensive, causing performance loss. Such losses depend on the monitoring mechanisms and characteristics of the monitored systems. A challenge in VMI-based SIEM systems is to select the appropriate VMI mechanisms, maximizing the usefulness of collected data and the detection and analysis capabilities while minimizing the performance impact on the monitored system.

In this abstract, we present our idea to enhance the functional DINGfest architecture [2] with elements that effectively help analysts (1) make decisions that balance between collecting rich security datasets while maintaining efficient systems, and (2) execute the decisions by adjusting the system. We add the following features:

- a control module for the VMI-based tracing tool
- a possibility to control or reconfigure the monitoring tool using an interactive visual interface
- a means to maintain user's awareness of the monitoring context, using a visualization of updated estimation and prediction of the performance impact of the deployed monitoring mechanisms

II. Current status

Fig. 1 shows the control module (a), the performance monitoring (b) and the details of the extended visual security analytics module (c) that we introduced to the DINGfest architecture in [2].

A. Configurable VMI

The data acquisition component of the DINGfest architecture (see Fig. 1) uses a library that can be configured to execute predefined sets of monitoring commands such as periodical extraction of process lists or continuous tracing of function and system calls. To monitor the running virtual machines (VMs), software breakpoints are inserted to trace the execution of processes [3].

In case of complex monitoring, this method can cause a large performance loss to the monitored VM. We extend the DINGfest VMI tool with a control module (Fig.1-(a)) which interprets and executes the control commands from the user (security analyst) in a dynamic way, i.e. during monitoring. The commands for reconfiguring the monitoring activities include the monitoring sets to be de-activated and activated along with their parameters, e.g., system call tracing and the list of system calls to trace. The control module is implemented as a consumer client that reads the commands published to a dedicated topic of the data streaming service.

B. Overhead measurements and cost prediction

The performance losses caused by VMI methods are not arbitrary and depend on the deployed analyses. A performance monitoring module (Fig.1-b) added to DINGfest architecture computes the overhead (cost estimation) and indicates the performance impact caused by reconfiguring the monitoring (prediction) in realtime. The module implements different methods to estimate the runtime overhead caused by the time during which a VM is interrupted with a breakpoint, e.g., *the sum of the time to retrieve information when an instruction of interest is reached and when it was executed and the total time for context switches from/to monitoring/monitored VM* divided by *the total computing time for running a program*. This allows to estimate the expected response time of a service, knowing its performance when run without monitoring.

When a reconfiguration command is received by the control module, its potential impact on the performance of the monitored VM is computed and this value will be shared to the user who can combine this information with their knowledge of the required datasets to understand the security situation, and therefore decide whether the reconfiguration should be executed or not. A challenge in this implementation is to find

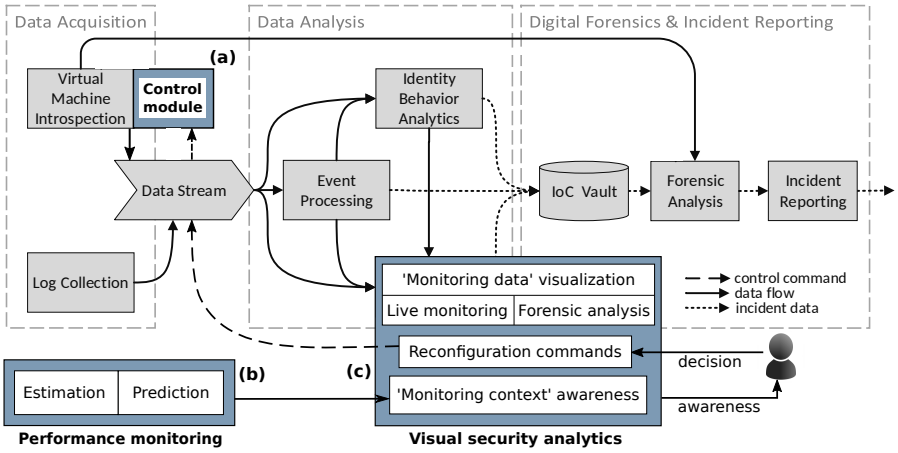

Fig. 1. Components of the DINGfest functional architecture (in light grey) [2] extended with components (in blue) that support the monitoring reconfiguration by the user.

the adequate measurements methods for a complex configuration in which the user combines different monitoring sets, e.g., system call tracing with repeating process list extraction.

C. Visual security analytics

The data analysis component in [2] features a data processing module and a visual security analytics that supports the investigation of security incidents, allows communicating with the storage and streaming services, and offers a continuous integration of the experts' domain knowledge into the automated analysis process by adjusting its parameters using the interactive visual interface. However, interaction of the user with the data acquisition is not supported yet. We extend the visual security analytics with elements that help the user make and apply reconfiguration decisions.

1) Visualization of monitoring data: It displays in realtime what is happening in the monitored VM, e.g., the system call activity [4], which is central to understanding what the applications are doing and identify the cause of the monitored VM's performance change.

2) Performance visualization: To wisely configure the monitoring sets, analysts need to be aware of the performance impact of currently active monitoring mechanisms, and the potential consequences of adding new ones. We visualize the evolution of the performance impact and update the estimation as often as needed in order to provide a realtime prediction and keep the analyst aware of the monitoring context, e.g., VM workload and active monitoring sets.

3) Reconfiguration command: It is used to build the commands for reconfiguring the VMI-based data acquisition component. It is implemented as a client producer publishing the commands to a specific topic in the streaming service to which the monitoring tool is listening.

III. Conclusion and Future work

Using VMI-based mechanisms for complex analyses in SIEM system can lead to large performance losses that are not always justified for the different threats to which the monitored environment is exposed. Security analysts need to understand the gradual trade-off between running complex monitoring for maximizing the data acquisition and maintaining the performance of the monitored resources. In this work in progress, we extend the DINGfest functional architecture [2] with elements that help analysts make such decisions.

The next step to our study consists of finding adequate methods to evaluate the effectiveness of the newly added elements in the architecture. We want to evaluate the accuracy of computed prediction by comparing the actual performance against the predicted ones in different execution contexts of the monitored VM and for different combination of monitoring sets. For instance, the currently used prediction algorithms rely on the work load of the monitored VM, therefore adequate approaches should be found which take into consideration the variation of activities at different times in order to preserve the accuracy of the cost prediction. We also plan to study the measurement of the performance impact when several VMI-based monitoring sets are combined by the user.

Acknowledgment

This research was supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest project (https://dingfest.ur.de), and by the Deutsche Forschungsgemeinschaft, as part of the ARADIA project.

References

[1] T. Garfinkel and M. Rosenblum, "A virtual machine introspection based architecture for intrusion detection," in *Proceedings of the Network and Distributed System Security Symposium, NDSS 2003*, 2003, pp. 191–206.

[2] F. Menges, F. Böhm, M. Vielberth, A. Puchta, B. Taubmann, N. Rakotondravony, and T. Latzo, "Introducing DINGfest: An architecture for next generation SIEM systems," in *SICHERHEIT 2018*, 2018.

[3] B. Taubmann, N. Rakotondravony, and H. P. Reiser, "CloudPhylactor: Harnessing mandatory access control for virtual machine introspection in cloud data centers," in *2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE, 2016, pp. 957–964.

[4] S. Laurén and V. Leppänen, "Virtual machine introspection based cloud monitoring platform," in *Proceedings of the 19th International Conference on Computer Systems and Technologies*. ACM, 2018, pp. 104–109.