SAIDS: A Self-Adaptable Intrusion Detection System for IaaS Clouds

Anna Giannakou, Louis Rilling, Christine Morin, Jean-Louis Pazat

SAIDS: A Self-Adaptable Intrusion Detection System for IaaS Clouds

Anna Giannakou*
LBNL, California, USA
agiannakou@lbl.gov

Louis Rilling
DGA, France
louis.rilling@irisa.fr

Christine Morin and Jean-Louis Pazat
Univ Rennes, Inria, CNRS, IRISA, France
firstname.lastname@irisa.fr

I. Introduction

IaaS clouds allow customers (called tenants) to deploy their IT as virtualized infrastructures. However IaaS clouds features, such as multi-tenancy and elasticity, generate new security vulnerabilities [1] for which the security monitoring must be partly run by the cloud provider to give visibility at the virtualization infrastructure level. Unfortunately the same IaaS clouds features make the virtualized infrastructures frequently reconfigurable and thus affect the ability of a provider-run security monitoring system to detect attacks [2]. This work addresses these issues for security monitoring systems based on off-the-shelf network intrusion detection systems (NIDSs) like Suricata [3].

a) Objectives: A security monitoring system tailored for IaaS clouds should address the following objectives:

1) self-adapt to infrastructure changes at the topology level (VM creation, deletion, migration and virtual network reconfigurations; physical server addition and removal), the traffic level (changes in the incoming and outgoing load), the application level (set of services deployed in the VMs and their updates);
2) enable tenants to customize the security monitoring using Service Level Agreements (SLAs);
3) scale with the monitoring load;
4) keep respecting security monitoring SLAs even during reconfigurations and introduce no new vulnerabilities in the provider's infrastructure.

b) Related work: Several approaches [4], [5], [6] propose Intrusion Detection Systems for cloud environments and partly satisfy Objectives 2 and 3 but not the self-adaptation objective 1. RemoteTrans [7] self-adapts only to application-level infrastructure changes and does not address Objective 3. VESPA [8], a self-protection monitoring architecture for IaaS clouds, addresses a different class of self adaptation based on security incidents but not Objective 1 and addresses neither multi-tenancy nor Objective 2. VMware NSX [9] addresses all objectives by allowing tenants to deploy a tightly-coupled NIDS (as well as other security appliances) for each VM or security group of VMs. However, similarly to [5], the resulting scalability is questionable because the amount of resources allocated to the NIDSs are not based on the monitoring load changes and thus should be over-provisioned.

*The work was done while the first author was working at Inria.

c) Contribution: In this context, we designed, implemented, and evaluated SAIDS, a self-adaptable NIDS framework for IaaS clouds based on off-the-shelf NIDSs and that addresses Objectives 1 to 4.

II. SAIDS

SAIDS brings the following features. (1) Probes detect the need for adaptation and SAIDS components are reconfigured accordingly. (2) Tenants can define service-specific NIDS rules in the SLA. (3) New NIDS instances are automatically deployed to scale with the traffic workload. Finally, (4) topology-level infrastructures changes are synchronized with SAIDS components reconfigurations to always monitor the network traffic seen by tenants' VMs according to the SLAs.

SAIDS consists of seven components depicted in Figure 1: the API, the Adaptation Manager (AM), the Infrastructure Monitoring Probes (IMP), the Local Intrusion Detection Sensors (LIDSs), the Adaptation Worker (AW), the Master Adaptation Driver (MAD), and the Mirror Worker (MW). The API extends the provider's API to allow tenants to write high-level descriptions of their monitoring requirements. These requirements are translated to SAIDS-specific adaptation arguments. The API definition is left for future work.

The LIDSs run on dedicated nodes under control of one MAD per node. A single node can host multiple LIDSs. An LIDS is composed of an AW and an off-the-shelf NIDS. Finally, we include one Mirror Worker per compute node. To synchronize the VM lifecycle with adaptation actions, SAIDS also features an optional safety mechanism on compute nodes.

The AM makes the adaptation decisions that affect the LIDSs upon the occurrence of dynamic events in the cloud infrastructure. Adaptation decisions are based on a monitoring strategy matching the SLAs and a maintained view of which LIDSs monitor each subset of VMs.

The IMPs detect topology changes (e.g. VM migration) and relate all necessary information to the AM (VM ID, internal and external IP address, port on the local switch, etc).

The LIDSs are responsible for analyzing network traffic that flows through subsets of local switches, depending on the monitoring strategy selected. The detection technique used can either be signature- or anomaly-based.

The AW has four roles: 1) makes the NIDS load its new configuration while keeping analyzing traffic as negotiated in

the SLA; 2) notifies the MAD upon completion of the adaptation process; 3) monitors the NIDS for failures and restarts it when necessary and 4) periodically reports LIDS-specific monitoring metrics (e.g. packet drop rate) to the MAD.

The MAD translates the adaptation parameters from the AM to LIDS-specific rules and reconfigures the traffic distribution on the local switch when a new LIDS is instantiated. The MAD also notifies the AM about performance degradation of existing LIDS(s). The MAD can handle multiple reconfiguration requests in parallel.

The MW guarantees that the traffic from a subset of VMs is correctly mirrored to the corresponding LIDS(s) node(s).

Finally, the safety mechanism prevents the VM that is involved in a dynamic event (e.g. a migrated VM) from resuming its execution before the adaptation process is completed. Tenants can choose to disable the safety mechanism.

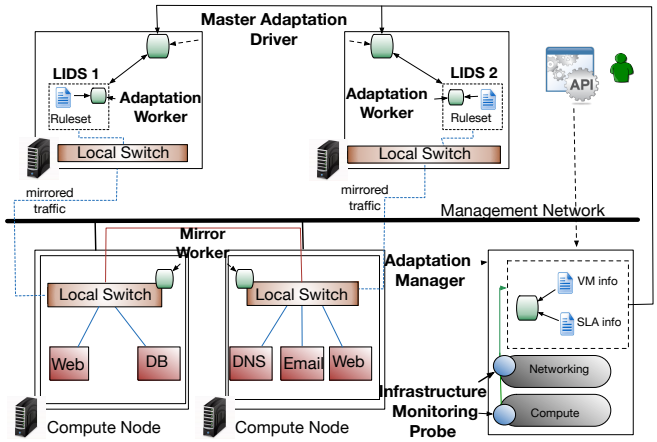

Fig. 1. SAIDS architecture

III. Evaluation

We evaluated the ability of SAIDS to guarantee adequate performance and minimized cost, for both tenants and the cloud provider, combined with a sufficient level of detection. We define the tenant-associated cost as the performance overhead in cloud hosted applications and the provider-associated cost as the time overhead in normal cloud operations and the increase in resource consumption.

We evaluated SAIDS performance in two aspects: adaptation speed and scalability. Adaptation speed refers to the time required for SAIDS to perform a full adaptation loop, from the time a dynamic event occurs until all involved LIDSs are successfully reconfigured. Scalability refers to the number of adaptation loops that SAIDS can handle in parallel and the overhead induced by parallelism in the reaction time of each phase of the adaptation loop. We studied scalability at the MAD and the AM levels. For the MAD level we evaluated the maximum number of LIDSs that a MAD can handle in parallel while for the AM level we evaluated the maximum number of MADs that an AM can handle in parallel.

The cost analysis refers to associated penalties on deploying SAIDS from the provider's and tenants standpoints. On the provider's side we calculated the overhead imposed by SAIDS to normal cloud operations (e.g. VM migration) while for the tenants we examined if SAIDS imposes any overhead in their applications performance.

Results obtained show that SAIDS imposes negligible overhead on normal cloud operations and that the performance overhead for tenants is also negligible. Since SAIDS LIDSs are passive monitoring devices that work on a copy of the VM-related traffic it does not impose any network overhead in the applications deployed in the monitored VM. Moreover, the traffic between different SAIDS components is using the management network, thus no additional traffic is created in the monitored network.

In our experiments, we measured that a single MAD located in a node with 24GB of RAM can handle up to 50 LIDSs, as each LIDS requires 460.1MB of RAM. We also showed that SAIDS is scalable with about 25% overhead in managing around 100 MADs and 5000 LIDSs compared to the fastest case of managing a single MAD and 50 LIDSs with no LIDS initially running. The memory capacity of our testbed limited the number of LIDSs managed in parallel to 5000, but when deploying SAIDS in a production cloud [10], where compute nodes have larger amounts of hardware resources, the number of LIDSs could significantly increase without expecting prohibitive performance overhead.

IV. Future Work

In the future we would like SAIDS to handle other types of IDSs, like host-based IDSs or network analyzers, and address failures in the reconfiguration of an LIDS. We plan to combine the security monitoring of tenants and the provider. As a medium term goal, we plan to implement the API used for expressing tenant monitoring requirements, address multi-tenancy and to include security events in the categories of adaptation sources.

References

[1] R. V. Deshmukh and K. K. Devadkar, "Understanding DDoS Attack & its Effect in Cloud Environment," in *Proc. ICAC3'15*, Jan. 2015.

[2] N. u. h. Shirazi, S. Simpson, A. K. Marnerides, M. Watson, A. Mauthe, and D. Hutchison, "Assessing the impact of intra-cloud live migration on anomaly detection," in *Proc. IEEE CloudNet'14*, Oct 2014.

[3] "Suricata Open Source IDS Engine," https://suricata-ids.org, accessed: 2015.

[4] S. Roschke, F. Cheng, and C. Meinel, "Intrusion Detection in the Cloud," in *Proc. IEEE DASC'09*, Dec 2009.

[5] C. Mazzariello, R. Bifulco, and R. Canonico, "Integrating a network IDS into an open source Cloud Computing environment," in *Proc. IAS'10*, Aug 2010.

[6] M. Ficco, L. Tasquier, and R. Aversa, "Intrusion Detection in Cloud Computing," in *Proc. 3PGCIC'13*, Oct 2013.

[7] K. Kourai and K. Juda, "Secure offloading of legacy idses using remote vm introspection in semi-trusted clouds," in *Proc. IEEE CLOUD'16*, June 2016.

[8] A. Wailly, M. Lacoste, and H. Debar, "VESPA: Multi-layered Self-protection for Cloud Resources," in *Proc. ICAC'12*, 2012.

[9] VMware NSX Technical Product Management Team, "VMware NSX for vSphere Network Virtualization Design Guide ver 3.0," VMware, Tech. Rep., Jun. 2017.

[10] "OVH Dedicated Servers," https://www.ovh.com/us/dedicated-servers/, accessed: 2017.