Self Adaptation in Security Monitoring for IaaS Clouds

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

Self Adaptation in Security Monitoring for IaaS Clouds

Anna Giannakou, Inria, IRISA
Louis Rilling, DGA
Frederic Majorczyk, DGA
Christine Morin, Inria, IRISA
Jean-Louis Pazat, INSA, IRISA

1. Introduction

Recent advances in the virtualisation area have made clouds a prominent solution for outsourcing information systems. Some of the key characteristics of cloud infrastructures include scalability, multitenancy and resource sharing. Tenants can create, destroy or reconfigure virtual resources with unprecedented ease. However, the same characteristics that make cloud environments agile and dynamic, also affect the ability of a security monitoring framework to successfully detect attacks [2] and sometimes introduce new security vulnerabilities that originate from both inside and outside the infrastructure. Traditional security monitoring solutions are not designed to automatically cope with reconfigurations of the virtual environment, and the potentially high rate of such reconfigurations makes it impossible for a security administrator to reconfigure the security monitoring setup accordingly. This requirement can potentially lead to misalignment between cloud administration and security administration workflows and ultimately to incomplete detection of threats. Moreover large scale security monitoring frameworks include various components (firewalls, IDSs, log collectors etc.) that perform different functionalities and are located in different areas or even outside the virtual infrastructure. For theses reasons, a successful cloud-tailored security monitoring infrastructure should be able to adapt its components based on changes in the infrastructure with little to no manual input.

1.1 Related Work

Providing a security monitoring framework for cloud environments has been the focus of several research projects. Some of them utilise virtual machine introspection techniques in order to gain a comprehensive and deep view of the monitored system [3][4]. However, these solutions often focus on intrusion detection in one particular virtual machine and are not able to adapt based on occuring changes.

In [1] Roschke *et al.* suggest to deploy intrusion detection sensors in various layers of the cloud model, those intrusion detection sensors being controlled by a central unit. The proposed solution is intrusive as it entails deployment of sensors inside client VMs.

VESPA [5], a self protecting monitoring architecture for IaaS clouds, addresses self adaptation but does not take into account multi-tenancy and per-tenant specific security requirements.

2. Thesis goal

Our goal throughout the duration of this thesis is to design and implement a self-adaptable security monitoring framework that is able to detect infrastructure related changes and automatically reconfigure its components accordingly. We have identified six major properties that our framework should comply to:

- Self adaptation: system components should be able to reconfigure themselves based on changes that occur in the cloud environment. The changes can be initiated either by the provider or the tenants. We identify three major change categories: hardware infrastructure related (i.e server addition or removal), virtual infrastructure related (VM creation, deletion, migration) and traffic related (fluctuation in the load of monitored traffic). The system should feature dedicated probes for detecting these high occurrence phenomena. Security related events such as detection of a DoS attack should also trigger reconfiguration of the involved components.
- Scalability: the system should instantiate new monitoring sensors in order to cope with variations in the volume of monitored traffic and

addition of new servers both at physical and virtual level.

- Customisation: the tenants should be allowed to customise the framework or parts of it for detecting specific attacks depending on the type of deployed services. The specific security requirements should be expressed in the Service Level Agreement.
- Isolation: the system should guarantee isolation of monitoring traffic for each tenant. Security components could share physical resources but separate per tenant instances should be deployed.
- Cost minimisation: a successful solution should achieve a high detection quality while maintaining the financial cost for both tenants and the provider at a minimal level. In order to achieve this objective the system should allow component sharing between tenants.
- Security: the system should address the aforementioned properties in a way that does not introduce novel security threats. The reconfiguration of security components or the instantiation of new ones, actions that are required in order to address self adaptation and scalability, should occur is such frequency that guarantees that they remain operational. Customisation of rules should guarantee that the newly applied ruleset does not exceed the capacity of the component dedicated to each tenant. Component sharing should maintain the same detection quality while achieving lower costs.

2.1 Threat model

We consider software attacks that originate both from virtual machines and outside the cloud infrastructure. Our system targets both service level (i.e SQL injection) and system level (directed towards the virtual or physical infrastructure) attacks.

3. Current status

In this context we have designed SAIDS, a self-adaptable intrusion detection framework that addresses the afore-mentioned objectives. As depicted in Figure 1 SAIDS consists of three major components that are run by the cloud administrator: the local Intrusion Detection Sensors (lIDS), the Infrastructure Monitoring Probes (IMP) and the Adaptation Manager (AM). The lIDSs are responsible for analysing network packets that flow through subsets of virtual switches using a signature based technique. We utilise containers for creating a separate lIDS instance per tenant in order to guarantee isolation. Alert correlation is out of the scope of this thesis. The IMPs are located inside the cloud controller and are responsible for detecting topology changes (for instance a vm migration) in the virtual infrastructure and relating all necessary information (identifier and IP of the VM that participates in the change, hostname of the physical host of the VM) to the AM. Finally the AM handles the reconfiguration of the lIDSs upon receiving a notification from the IMP. The AM relates the incoming information to a list of services running in the VM and selects the set of additional rules than need to be activated in the lIDS responsible for the virtual switch at the VM's new location. This set of additional rules also include rules that address a tenant's specific security requirements as stated in the Service Level Agreement. The AM also decides whether to deploy a new lIDS depending on the traffic load. The communication between the AM and the lIDSs is handled through a secure channel.

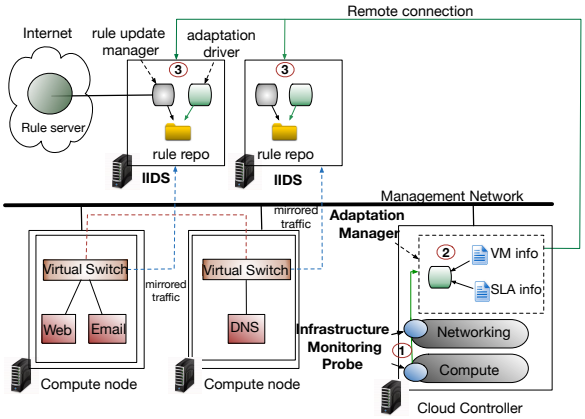

Figure 1. The SAIDS architecture

We have conducted an early qualitative evaluation of SAIDS based on a scenario that manifests the need for adaptation. In our scenario we demonstrate the security gap introduced when a

migration of a VM occurs and how we handle it by adapting the lIDS responsible for monitoring the traffic in the new location of the VM. The overall migration time with the adaptation process is 4s while without adaptation it takes 2.1s. Although the adaptation process is executed in parallel with the migration, we observe that reconfiguring an lIDS is increasing significantly the overall time.

4. Future Work

In the future we plan to combine the security monitoring of tenants and provider infrastructure in order to achieve cost minimization. Next steps include giving partial control of the monitoring framework to tenants and expanding our architecture by including other types of devices such as log collectors, firewalls and aggregators. We also intend to address scalability issues.

References

[1] S. Roschke et al. Intrusion Detection in the Cloud. In *Proc. DASC*, 2009.

[2] N. Shirazi et al. Assessing the impact of intra-cloud live migration on anomaly detection. In *Proc. CloudNet*, 2014.

[3] Tal Garfinkel and Mendel Rosenblum. A Virtual Machine Introspection Based Architecture for Intrusion Detection In *Proc. NDSS*, 2003.

[4] Joshi, Ashlesha and King, Samuel T. and Dunlap, George W. and Chen, Peter M. Detecting Past and Present Intrusions Through Vulnerability-specific Predicates. In *Proc. SOSP*, 2005.

[5] A. Wailly et al. VESPA: multi-layered self-protection for cloud resources. In *Proc. ICAC*, 2012.