Multi-cloud applications security monitoring

Pamela Carvallo[1,2], Ana R. Cavalli[1,2], Wissam Mallouli[2], and Erkuden Rios[3]

[1] SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay, Évry, France
[2] Montimage, Paris, France
{pamela.carvallo,ana.cavalli,wissam.mallouli}@montimage.com
[3] Tecnalia, Derio, Spain,
erkuden.rios@tecnalia.com

Abstract. The issue of data security and privacy in multi-cloud based environments requires different solutions for implementing and enforcing security policies. In these environments, many security aspects must be faced, such as security-by-design, risk management, data privacy and isolation, and vulnerability scans. Moreover, it also becomes necessary to have a system that interrelates and operates all security controls which are configured and executed independently on each component of the application (service) being secured and monitored. In addition, thanks to the large diffusion of cloud computing systems, new attacks are emerging, so threat detection systems play a key role in the security schemes, identifying possible attacks. These systems handle an enormous volume of information as they detect unknown malicious activities by monitoring different events from different points of observation, as well as adapting to new attack strategies and considering techniques to detect malicious behaviors and react accordingly.

To target this issue, we propose in the context of the MUSA EU Horizon 2020 project [1], a security assurance platform that allows monitoring the multi-cloud application deployed in different Cloud Server Providers (CSPs). It detects potential deviations from security Server Level Agreements[4] (SLAs) and triggers countermeasures to enforce security during application runtime.

Keywords: Cloud computing, Security monitoring, Service Level Agreement, Detection.

1 Introduction

In this section we present the main challenges for monitoring of multi-cloud environments and the related work on threat detection systems. We also present in this section the paper organization.

[4] A formal, negotiated document that defines in quantitative and qualitative terms the service being offered to a Cloud Service Client (CSC). For more information see [8, 17].

1.1 Monitoring Challenges in multi-cloud environments

Monitoring is a solution that is required to control the correct operation of the whole system running in a multi-cloud environment. According to the taxonomy proposed by [13] and [12], the term multi-cloud denotes situations were a consumer (human or service) uses multiple independent clouds, unlike Cloud Federations that are achieved when a set of cloud providers voluntarily interconnect their infrastructures to allow sharing of resources among them. A few concrete multi-cloud solutions exist, addressed in research projects like MUSA, OPTIMIS, mOSAIC, MODAClouds, PaaSAge and Cloud4SOA [6, 11]. It is out of the scope of this paper to offer a complete survey of such activities. We suggest the interested reader the following works: [13, 5, 19].

Malfunctioning or even minor problems in a Virtual Machine (VM) could introduce vulnerabilities and instability to other VMs, as well as the integrity of the host machine. In this paper, the monitoring function is needed to be able to precisely understand what is going on in the network, system and application levels, with a twofold objective. First, it is necessary for improving the security in the communications and services offered by the multi-cloud virtual environments. Second, from the administration and managements point of view, it will help ensure the environments health and guarantee that the system functions as expected and respects its security SLA.

Existing monitoring solutions to assess security can still be used in virtualized network environments. Nevertheless, they need to be adapted and correctly controlled since they were meant mostly for physical and not virtual systems and boundaries, and do not allow fine-grained analysis adapted to the needs of CSCs and virtualized networks. The lack of visibility and controls on internal virtual networks, and the heterogeneity of devices used make some performance assessment applications ineffective. On one hand, the impact of virtualization on these technologies needs to be assessed. For instance, Quality of Service (QoS) monitoring applications need to be able to monitor virtual connections. On the other hand, these technologies need to cope with ever-changing contexts and trade-offs between the monitoring costs and the benefits involved.

Tools such as Ceilometer [2], a monitoring solution for OpenStack, provide efficient collection of metering data in terms of CPU and network costs. However, it is focused on creating a unique contact point for billing systems to acquire all of the measurements they need, and it is not oriented to perform any action to try to improve the metrics that it monitors. StackTach [4] is another example oriented to billing issues that monitors performance and audits the OpenStacks Nova component. Similarly, but not specifically oriented to billing Collectd [9] gathers system performance statistics and provides mechanisms to store the collected values. A recent project from OPNFV named Doctor [3], focuses on the creation of a fault management and maintenance framework for high availability of network services on top of virtualized infrastructures. All the mentioned solutions do not consider their monitoring functionality to tackle security issues.

In terms of security, OpenStack provides a security guide [14] providing best practices determined by cloud operators when deploying their OpenStack so-

lutions. Some tools go deeper in order to guarantee certain security aspects in OpenStack, for instance: Bandit [16] provides a framework for performing security analysis of Python source code while Consul [10] is a monitoring tool oriented to service discovery that also performs health checking to prevent routing requests to unhealthy hosts.

1.2 Related work on threat detection systems

Intrusion Detection Systems (IDS) in cloud-based environments usually correspond to a hardware device or software application that monitors activity (e.g. network, host, user) for malicious policy violations. Zbakh et al. evaluated in [18] several IDS architectures through proposed multi-criteria decision technique, according to the above introduced requirement together with few others such as:

- Performance
- Availability
- Scalability
- Secure and encrypted communication channels
- Transparency with respect to end-users
- Information Security Policies as input to the architecture
- Accuracy, including the number of false positives (FP), false negatives (FN)
- Detection methods used

According to such literature, IDS architectures may vary if they are distributed, centralized, agent-based or collaborative [18]. Patel et al. [15] provided an extended systematic-based study of intrusion detection systems, presenting a classification with regards to response time, alarm management, detection method, data collection type, among others. In general, these systems are designed with the following modules: data collection (Section 2.3) and preparation (Section 3.1) are performed through a sensor or existing database which works as an input for the data analysis and detection (Section 3.2). The latter engine corresponds to the module of the algorithms implemented to detect suspicious activities and known attack patterns.

In the context of this paper, we consider the monitoring of multi-cloud based application where each application component can be deployed in a different cloud service provider. This architecture brings more challenges to be able to fulfill an end-to-end security monitoring of the application execution and communication at runtime. To our knowledge, no security monitoring solution has been designed for such multi-cloud distributed systems. The main contribution of this paper is the design and development of a security assurance platform that provides an answer to these challenges.

1.3 Paper organization

The paper is organized as follows. In section 2, we present an overview of the multi-cloud security assurance platform, which is part of the approach developed

in the MUSA project. The platform is composed by several modules that are described in detail. Section 3 presents the workflow implemented in this platform. Section 4 summarizes and gives some elements for discussion of the presented work. Finally, section 5 gives the conclusion of this work.

2 The MUSA security assurance platform SaaS

2.1 The MUSA framework

The main goal of MUSA is to support the security-intelligent life-cycle management of distributed applications over heterogeneous cloud resources, through a security framework that includes: a) security-by-design mechanisms to allow application self-protection at runtime, and b) methods and tools for the integrated security assurance in both the engineering and operation of multi-cloud applications. MUSA overall concept is depicted in the figure below.

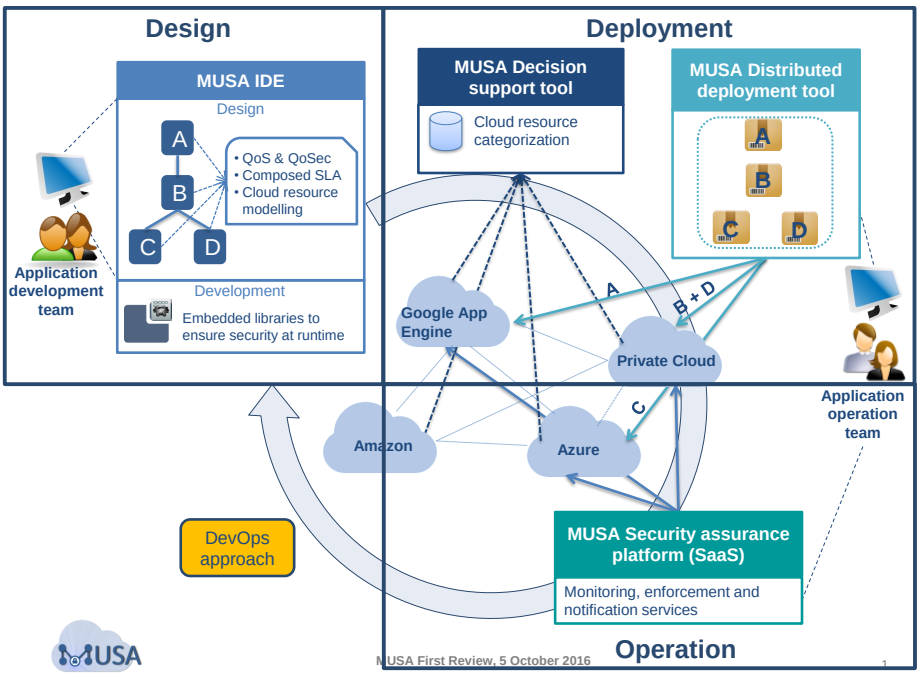

Fig. 1. MUSA overall concept

MUSA framework combines 1) a preventive security approach, promoting security-by-design practices in the development and embedding security mechanisms in the application, and 2) a reactive security approach, monitoring application runtime to mitigate security incidents, so multi-cloud application providers

can be informed and react to them without losing end-user trust in the multi-cloud application. An integrated coordination of all phases in the application life-cycle management is needed in order to ensure the preventive oriented security to be embedded and aligned with reactive security measures.

MUSA focuses security from with reactive approach, where we designed and implemented a security assurance platform deployed as a service. This service is available following this link http://assurance-platform.musa-project.eu/ and a demonstration of the tool is available on YouTube following this link: https://www.youtube.com/watch?v=zc6p-0H9yFo.

2.2 The MUSA Security Assurance Platform overview

The MUSA Security Assurance Platform (MUSA SAP) fits the operation phase of the MUSA framework and it is devoted to continuously monitor and analyze multi-cloud application security with the possibility of activating automatic reactions (based on security enforcement libraries) and sending notifications (alerts and violation information) in case of detecting security issues with the ultimate objective of maintaining confidentiality and privacy of sensitive data and communications. The MUSA SAP needs four main inputs to work correctly:

- The Security SLA of the application to monitor: The MUSA SAP recuperates the single application components SLAs or the multi-cloud composite application SLA (shown in Figure 1 in the Design phase). The latter refers to the final SLA composed out of the security policies of each of the individual services and their specific SLAs, therefore resulting in a single SLA associated to the whole multi-cloud implementation. From that input, the MUSA SAP can monitor the security of single components and from composite SLA, it can check the end-to-end security of the multi-cloud application taking the communication exchanges between remote components into account.
- The application deployment plan: From this plan, the MUSA SAP recuperates the list of monitoring agents deployed with each application component as well as their IP addresses. This information is very important to link the monitoring agent with the application component in order to monitor the right security metrics that are specified in the application component security SLA.
- The monitoring agents: The MUSA SAP configures the monitoring agents in order to measure the security metrics related the security controls required for an application component (and specified in their security SLA). The reported measurements and events are correlated in the platform SaaS to detect potential alerts or violations.
- The enforcement agents: The MUSA SAP activates enforcement agents in case of security issue detection with the ultimate objective of maintaining confidentiality and privacy of sensitive data and communications.

2.3 MUSA SAP Monitoring agents

To be able to deeply analyze security, the MUSA SAP relies on different agents to be installed in different VMs or containers where application components are deployed. These agents collect data coming from network, system and application internals and send them to the monitoring platform MUSA SAP. Among these agents, we have:

Network monitoring agent This is a monitoring solution that combines a set of functionalities presented in the following list:

- Data capture, filtering and storage;
- Events extraction and statistics collection; and
- Traffic analysis and reporting providing, network, application, flow and user level visibility.

Through its real-time and historical data gathering, the network monitoring agent facilitates network performance monitoring and operation troubleshooting. With its advanced rules engine, the monitoring agent can correlate network events in order to detect performance, operational, and security incidents.

System monitoring agent The System agent monitors system resources which may be the cause of server performance degradation and spots performance bottlenecks early on. The System agent relies on Linux `top` command which is used frequently by many system administrators to monitor Linux performance and it is available under many Linux/Unix like operating systems. The top command used to display all the running and active real-time processes in an ordered list and updates it regularly. It displays CPU usage, Memory usage, Swap Memory, Cache Size, Buffer Size, Process PID, User, Commands, among others. It also shows high memory and CPU utilization of all running processes.

Application monitoring agent The role of the Application agent is to deliver information about the internal state of the multi-cloud application component to the MUSA SAP during its operation. It continuously checks and monitors application health. It notifies the MUSA SAP about measurements of execution details and other internal conditions of the application component. The application monitoring agent is a Java library built of two parts. The first part is an aspect to be weaved into the application code via point-cuts in order to send application-internal tracing information to the MUSA security assurance platform for analysis. It is composed of a set of functions that can be weaved in strategic application points to capture relevant internal data. The second part connects the aspect with the notification tool via a connector library and it provides a simple interface to send log data to the MUSA SAP in a secure way. In other words, the application monitoring agent is responsible for extracting the information from the application environment, and the connector is responsible for transferring it.

2.4 MUSA SAP Enforcement agents

Prevention, monitoring, detection, and mitigation generally illustrate the defense life-cycle. Prevention involves the implementation of a set of defenses, practices, and configurations prior to any kind of attack, with the aim of reducing the impact of such attack. These issues could be addressed by network security, data protection, virtualization and isolation of resources. Traditionally, well-known countermeasures have focused on dealing with threats through a variety of methods devised around questions such as where is the attack detected? How is the attack detected? What is the response mechanism? Where to apply the response mechanism? Where is the control (decision) center from which filtering rules are taken? Previous studies have assessed the analysis of such mechanisms, for instance, Carlin et al. [7] studied vulnerabilities and countermeasures and proposed a flow chart showing the exiting DDoS cloud protection systems and comparing the implementation of different features in the proposed systems. Other methods utilized are profiling based techniques, in order to discriminate the mis-usability from users (e.g. trying to gain privileges); IDS, pattern matching in the search for specific confidential words trying to be breached, or queries in databases monitoring. The MUSA SAP integrates a set of security enforcement agents that can be easily deployed when a security breach is detected. As an example, a high availability framework to ensure application availability even under charge.

3 The MUSA SAP workflow

The MUSA workflow is illustrated in Figure 2 and it's composed of four main modules followed by the gathering of data from different monitoring agents. More details about these steps are provided in the next subsections.

3.1 Preprocessing the data

This module has a particular challenge, which is extracting the right information from the collected data collected by different monitoring agents and from different CSPs, in order to build the correct usage profiles. This unit is meant to be dynamic, where features are analyzed in regard of time-based contextual information. This has the advantage of decreasing the usage of resources for the analysis of large amounts of data, therefore increasing the performance of the framework and reasoning detection. Also following this direction, it is relevant at the moment of keeping a non-redundant dataset. Additionally, in real cloud environments, periodic reports may be subject to loss or high latency, due to the applications elasticity or VM-related features (e.g. restarting a VM, rolling back.). Hence, it is relevant to be able to be resilient to the lack of all types of log information at all times and be able to construct the *possible* missing pieces by studying the whole picture of what type of features are being received and how to treat them to build the best profile of the database status.

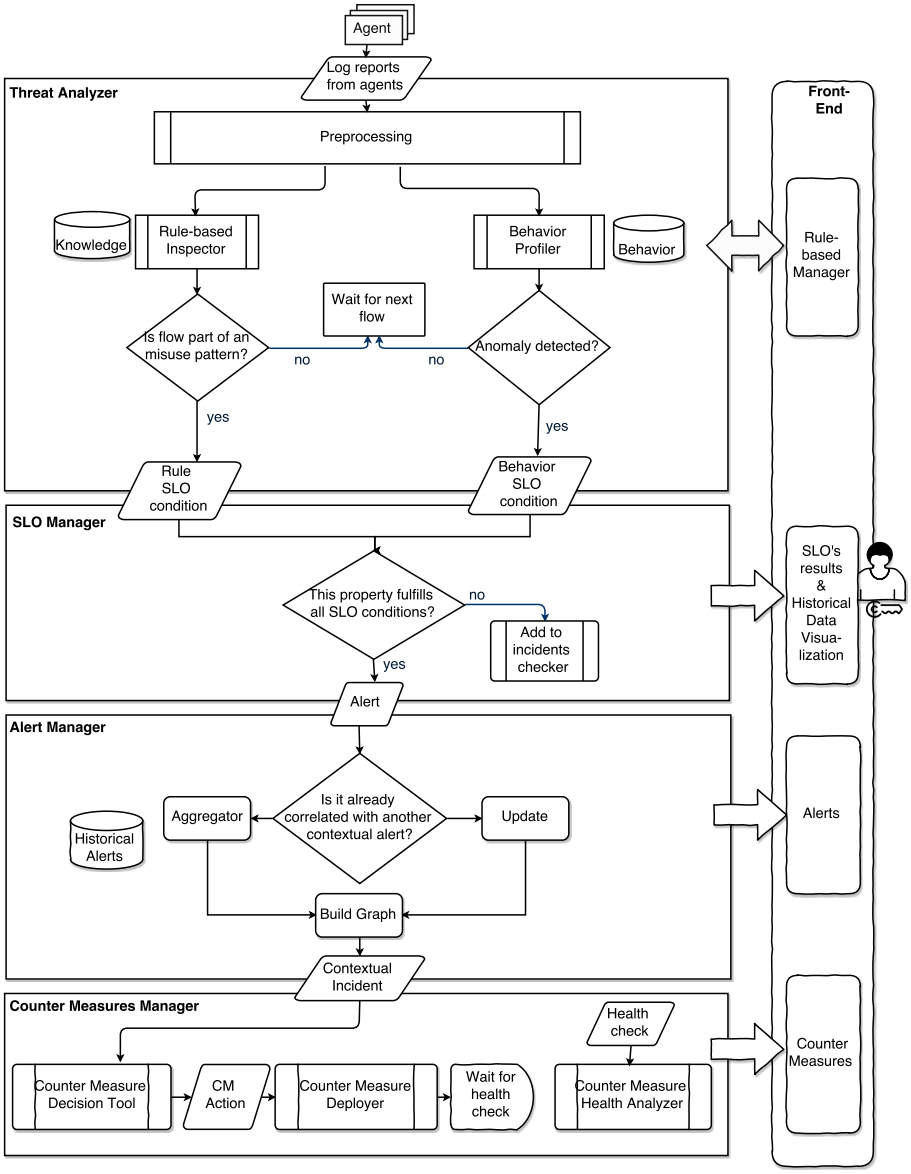

Fig. 2. The MUSA SAP workflow

3.2 Detecting threats and anomalies

The Threat Analyzer module consists in two sub-modules: Rule-based inspector and a Behavior profiler, as shown in Figure 2.

The first resides in an engine that receives information events from the prepossessing module, regarding users accessing to the sensitive data and they are checked against these permission rules. Additionally to this policy control, some of the attributes obtained from the agents, are inspected for specific pattern-matching detection.

The second module also receives the preprocessed data and comprehends two functions: the online-learning and the anomalies detection.

Most of the literature related to anomaly detection, establishes a separated two-stage process where systems are trained with *normal* data for second-stage comparison with new incoming information. This idea lacks of dynamism, as cloud behavior may vary in mid-long term, and is highly dependent of the nature of the training data. Therefore, we propose a self-learning module which is able to feed and update itself dynamically from new data flows. This system will discriminate if it is appropriate to self-feed itself or not, lowering the possibility of training the engine with *malicious activity* as *normal*.

The model uses a semi-supervised learning, given the fact that new input data has no a priori labeling and needs to be classified on the basis of their statistical properties only. The supervised component comprehends a smaller labeled dataset created in a lab environment, which learns from known attacks.

3.3 Service Level Objectives (SLO) Manager

The SLO Manager is able to check measured attributes we need to assert which objectives are useful in defining an anomalous behavior or a disrespected rule. The latter is already paved since it consists in rules that are continuously checked.

3.4 Alert Manager and Countermeasures Manager

The correspondent modules *Alert Manager* and *Countermeasure Manager* from Figure 2, respond to a policy based of existing alert and countermeasure mechanisms, given the severity of the incident diagnosed. The last module is intended to advise the CSPs and may consist in notifying the administrator rolling back the composite application, replicating database, upgrading passwords complexity, disabling specific user, among others. The latter presents a crucial challenge because sometimes CSPs are unaware precisely of the countermeasures to consider, because there are no established relationships among cloud components and their dependencies. This can be solved by clarifying these relationships as it is currently being done in MUSA project where a multi-cloud composition model is being defined.

4 Discussion

The MUSA SAP is proposed as a service that needs to be deployed in the suitable CSP (or CSPs since we can divide the platform into independent components

or microservices), offering security controls according to the application needs (including security requirements).

Starting from this statement, the MUSA framework can be applied in the design, SLA generation, CSP selection and deployment phases in order to better select the CSP that fulfills its requirements (including robustness against attacks). Moreover, the MUSA SAP is able to enforce the security by executing the necessary countermeasures to repel security issues or to mitigate their undesired effects.

Its real-time data collection and analysis, together with its virtualized nature, makes the MUSA SAP a strong prototype ready to be used in industrial environments. It offers multi-cloud application developers a cloud-based integrated tool to perform real-time, SLA-based, end-to-end security monitoring and enforcement. With the definition of personalized security SLAs, the automatic deployment of the monitoring probes, together with the virtualized operation, the multi-cloud application developers just need to specify their security requirements and the MUSA SAP will do the rest: monitoring heterogeneous sources (network, application, containers, etc.), analyzing the collected information in real-time, enforcing the security, and offering detailed visual reports to keep CSCs aware of their systems health.

5 Conclusion

In this paper, we have presented a security assurance platform for multi-cloud applications. This platform includes techniques to perform the monitoring of these applications that are deployed over heterogeneous cloud resources. This platform is also based on the concept of security SLA in order to detect potential deviations of security rules and trigger countermeasures to protect applications against attacks. The proposed framework presents several advantages: providing preventive security based on the use of security-by-design practices in the applications' development. It also guarantees applications protection by countermeasures techniques to mitigate security incidents and providing applications with reaction mechanisms. The MUSA SAP is being evaluated in the context of two industrial case studies: one related to *smart cities* and the other related to the transport industry. The preliminary results are positives and will be a subject of complementary publication.

Acknowledgment

The project leading to this paper has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No. 644429.

References

1. MUSA Project. http://www.musa-project.eu/, (Retrieved January 2017)

2. Openstack ceilometer. http://docs.openstack.org/developer/ceilometer/, (Retrieved January 2017)
3. OPNFV Doctor. http://wiki.opnfv.org/doctor, (Retrieved January 2017)
4. Stacktach. http://stacktach.readthedocs.org/en/latest/index.html, (Retrieved January 2017)
5. Lifecycle management of service-based applications on multi-clouds: a research roadmap (2013)
6. Multi-Cloud: expectations and current approaches (2013)
7. Carlin, A., Hammoudeh, M., Aldabbas, O.: Intrusion Detection and Countermeasure of Virtual Cloud Systems - State of the Art and Current Challenges. International Journal of Advanced Computer Science and Applications 6(6) (2015)
8. Casola, V., Benedictis, A.D., Rak, M., Rios, E.: Security-by-design in clouds: A security-sla driven methodology to build secure cloud applications. Procedia Computer Science 97, 53 – 62 (2016), http://www.sciencedirect.com/science/article/pii/S1877050916320968, 2nd International Conference on Cloud Forward: From Distributed to Complete Computing
9. Collectd: http://collectd.org/, (Retrieved January 2017)
10. Consul: https://www.consul.io/, (Retrieved January 2017)
11. Ferry, N., Rossini, A., Chauvel, F., Morin, B.: Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems. 2013 IEEE Sixth International Conference on Cloud Computing (2013)
12. Global Inter-cloud Technology Forum: Use Cases and Functional Requirements for Inter-Cloud Computing. Tech. rep. (2010)
13. Grozev, N., Buyya, R.: Inter-Cloud architectures and application brokering: taxonomy and survey. Software - Practice and Experience 44(3), 369–390 (2012)
14. Guide, O.S.: http://docs.openstack.org/sec/, (Retrieved January 2017)
15. Patel, A., Taghavi, M., Bakhtiyari, K., Celestino Júnior, J.: An intrusion detection and prevention system in cloud computing: A systematic review. Journal of Network and Computer Applications 36(1), 25–41 (Jan 2013)
16. Project, B.: http://wiki.openstack.org/wiki/Security/Projects/Bandit, (Retrieved January 2017)
17. Rios, E., Mallouli, W., Rak, M., Casola, V., Ortiz, A.M.: SLA-Driven Monitoring of Multi-cloud Application Components Using the MUSA Framework. ICDCS Workshops (2016)
18. Zbakh, M., Elmahdi, K., Cherkaoui, R., Enniari, S.: A multi-criteria analysis of intrusion detection architectures in cloud environments. In: 2015 International Conference on Cloud Technologies and Applications (CloudTech). pp. 1–9. IEEE (2015)
19. Zeginis, C., Kritikos, K., Garefalakis, P., Konsolaki, K.: Towards cross-layer monitoring of multi-cloud service-based applications. Lecture Notes in Computer Science pp. 188–195 (2013)