SLA-Based Continuous Security Assurance in Multi-Cloud DevOps

Erkuden Rios[1], Massimiliano Rak[2], Eider Iturbe[1], and Wissam Mallouli[3]

[1] Fundación Tecnalia Research & Innovation. Derio, Spain
[2] University of Campania Studies Luigi Vanvitelli, Naples, Italy
[3] Montimage Research & Development. Paris, France

Abstract. Multi-cloud applications, i.e. those that are deployed over multiple independent Cloud providers, pose a number of challenges to the security-aware development and operation. Security assurance in such applications is hard due to the lack of insights of security controls applied by Cloud providers and the need of controlling the security levels of all the components and layers at a time. This paper presents the MUSA approach to Service Level Agreement (SLA)-based continuous security assurance in multi-cloud applications. The paper details the proposed model for capturing the security controls in the offered application Security SLA and the approach to continuously monitor and asses the controls at operation phase. This new approach enables to easily align development security requirements with controls monitored at operation as well as early react at operation to any possible security incident or SLA violation.

Keywords: multi-cloud security, multi-cloud monitoring, security assurance, security in DevOps, cloud security controls, cloud security SLA.

1 Introduction

Despite the tremendous growth in the expansion of Cloud Computing [1], security is still one of the major drawbacks of Cloud adoption [2]. Many consumers continue to be reluctant to go into the Cloud due to the lack of transparency and control over the security features the Cloud Service Providers offer. A growing trend to ease transparency consists in the use of cloud Security Service Level Agreements (SLAs). A cloud SLA is a contractual agreement between the Cloud Service Provider (CSP) and the Cloud Service Customer (CSC) specifying the security grants offered by the consumed cloud service [3].

In: M.G. Jaatun, D.S. Cruzes (eds.): Proceedings of the International Workshop on Secure Software Engineering in DevOps and Agile Development (SecSE 2017), published at http://ceur-ws.org

The research on cloud SLAs started some years ago and a collection of outcomes from initial EU-funded projects on the subject can be found in [4]. Since then, recent projects like SPECS [5], SLALOM [6] and SLA-READY [7] have enlarged the cloud SLA solution oriented results and provided relevant advances in tools, legal transparency and reference models rising understanding of non-security expert consumers, including SMEs, on negotiated service level clauses.

Nevertheless, several challenges for Security assurance in multi-cloud remain, like those tackled in this work. One one hand, we deal with the automatic creation of the (multi-)cloud applications' Security SLAs taking into account the SLAs of the composing components and the cloud services they use. And on the other hand we provide a solution for the continuous monitoring of such Security SLA, considering controls in all the layers involved: cloud service provider, system, network, application.

This paper presents the MUSA solution to SLA-based security assurance for multi-cloud applications that use or have their components deployed in distributed cloud services. The solution is based on the MUSA DevOps approach [8] that allows for the security-aware development and operation of such applications considering security as one of the major drivers in application life-cycle. The solution not only enables the automatic creation of the offered Security SLA of the multi-cloud application, but it also enables to monitor at runtime the security service level objectives specified in the SLA. The solution is part of the MUSA framework developed within the European Union's H2020 research project MUSA [9].

The MUSA framework is a DevOps oriented solution that seamlessly integrates different tools to support multi-disciplinary DevOps teams in the security-aware life-cycle of (multi-)cloud applications, from application design (including its Security SLA creation) till continuous assurance at operation.

The paper is structured as follows. After the introductory section, the Section 2 describes the state of the art in security SLAs for multi-cloud based applications. The Section 3 introduces the complete MUSA workflow and framework for the security-intelligent life-cycle management of multi-cloud applications. In Section 4 we detail the proposed approach to Security SLA Assurance in multi-cloud, which is part of the MUSA DevOps framework. The Section 5 describes the validation of the solution in two use cases in the domains of flight scheduling prototypes and smart mobility services. Finally, the Section 6 presents the conclusions and the future work.

2 Security SLAs in multi-cloud

The term *inter-cloud computing* was defined in [10] as: *A cloud model that, for the purpose of guaranteeing service quality, such as the performance and availability of each service, allows on-demand reassignment of resources and transfer of workload through a interworking of cloud systems of different cloud providers based on coordination of each consumers requirements for service quality with each providers SLA and use of standard interfaces.*

Therefore, *inter-cloud* indicates the usage of cloud resources from multiple Cloud Service Providers (CSPs). These multiple CSPs can be part of a *cloud federation*, i.e. voluntarily collaborate to allow sharing of cloud resources between them, or they can be independent, with no need to exist an explicit collaboration agreement or interconnection among them.

In this work we focus on the security assurance of *multi-cloud applications* which components use or are deployed in multiple cloud services offered by independent CSPs, which could be heterogeneous. Therefore, they pose more challenges to the design, creation and monitoring of their security SLAs because they combine services and security mechanisms from diverse and independent sources.

In order to be able to offer a holistic assurance in multi-cloud applications we need: i) mechanisms for SLA creation focused on security properties to derive composite Security SLA of applications that combine multiple clouds, and ii) mechanisms for continuous monitoring of such Security SLAs taking into account that heterogeneous controls can be applied by multiple providers at different layers: application, system or network.

2.1 The Security SLA model

According to ISO/IEC 20000-1 [11] a Service Level Agreement (SLA) is a documented agreement between the service provider and customer that identifies services and service level objectives.

When it comes to cloud, *a Cloud SLA is a contract framework that defines the terms and conditions necessary to fulfil the obligations of a Cloud Service Provider (CSP) for the service(s) offered to a Cloud Service Consumer (CSC)* [12]. In this line, the cloud Security SLA enables to express the security policy associated to cloud services offered to CSCs. The cloud Security SLA declares the set of Service Level Objectives (SLOs) related to security aspects of the service in terms of thresholds on well defined security metrics that relate to security controls of the service.

As defined by NIST Security Control Framework [13], security controls are *safeguards or countermeasures prescribed for an information system or an organization designed to protect the confidentiality, integrity, and availability of its information and to meet a set of defined security requirements.*

The NIST Control Framework [13] that is adopted in our approach gathers more than 900 security controls, identified by a unique identifier and a name, and are organised in families, such as Access Control (AC), Identification and Authentication (IA), Risk Assessment (RA) , System and Communications Protection (SC), System and Information Integrity (SI), etc.

Other security control frameworks for cloud exist such as Cloud Security Alliance's Cloud Control Matrix [14] and ISO/IEC 27017 [15], but NIST offers greater maturity, richness and granularity of the controls.

In our approach, we adopt the SPECS cloud Security SLA model, described in detail in [16] and based on the WS-Agreement standard [17]. This model has been extended with provider-specific information and security-related concepts

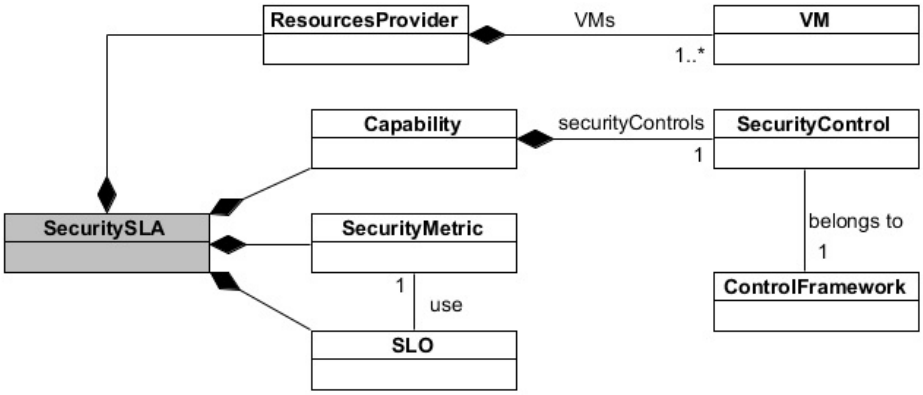

Fig. 1. Security SLA model from SPECS.

and its use adapted to multi-cloud environments. An extract of the model with the main concepts is shown in Fig.1.

As it can be seen in the model, *Security capability* concept refers to a grouping of security controls. Security capabilities are defined by NIST [13] as *sets of mutually reinforcing security controls*. Capabilities and controls are enforced by means of suitable software and/or hardware mechanisms, which are deployed by the resources provider either on the resources provided to the customer or on external resources. The Service Level Objectives use *security metrics* to express the target level that is guaranteed in the SLA. The security metrics that can be expressed by using the MUSA SLA Generator are part of the security metric catalogue presented in [18].

2.2 The Security SLA composition

The goal of Security SLA composition is to identify the security policy of each application component and the security policy of the overall application, i.e. the set of security controls that can be declared in the composite application Security SLA.

State of the art techniques of SLA composition range from ontology based [19] to WS-Agreement based [20], but all are focused on reliability and performance controls, and therefore, they are hardly reusable in security context.

In multi-cloud based applications, the Security SLA that can be offered to the application customers depends on how the components are deployed, the relationships among them, the number and type of the cloud resources they use and the Security SLAs of all the individual parts.

In our solution we rely on the approach detailed in [21] for obtaining the multi-cloud application composite Security SLA. This is a novel technique that enables to compute and declare in terms of standard security controls the security policy of applications that orchestrate multiple cloud services. The technique relies on graph-based models that synthesize the deployment architecture of

the distributed application, the relationships among the services composing the application (e.g. uses, is deployed in, protects, etc.), the SLA Templates of the services and the SLAs declared by the CSPs in use.

The SLA Template (SLAT) of the service is defined as *the document that describe the security policy implemented by a service according to its internal configuration and not taking into account the effect of deployments.* Therefore, Security SLA templates define the required security Service Level Objectives (SLOs) of the components in the basis of the SLOs required over the cloud services they will use.

In our approach the SLATs of the application components are obtained in a two-step process: in the first step, a risks analysis process is performed where each threat over the component is associated to the relevant security controls that minimize the risk; in the second step, each control required by the risk analysis is assessed against the code of the component with the aid of a dedicated questionnaire. These controls are those that would like to be granted by the component and they could be: implemented by the component, required on the cloud service it uses, or requested to MUSA security agents, as explained later.

The proposed SLA composition process translates each SLATs associated to the components to a SLA. The composition is performed on a per security control basis and it assumes the controls can be independently evaluated.

3 Security Assurance in multi-cloud

Even if not particularly oriented to multi-cloud, there exist a number of Cloud systems monitoring solutions such as those collected in the surveys provided in [22] and [23]. The state of the art multi-cloud monitoring solutions, e.g. from MOSAIC project [24] and PaaSage project [25], SeaClouds project [26] mainly focus on elasticity policies and quality of service (QoS) but lack specific or rich support to security control. This is the case of the framework offered in PaaSage for model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems that relies in the enactment of the application model (written in CloudML language [27]). In this framework monitoring is dependent on the definition of the metrics in CloudML language in the application model, rather than in an standard Service Level Agreement (SLA) format (such as Web Service Agreement), which limits the approach.

Initiating the path towards security assurance, the CUMULUS project [28] delivered an integrated framework of models, processes and tools supporting the certification of security properties of cloud services (IaaS, PaaS or SaaS).

On security specifics, the SPECS project [5] delivered an open source framework to offer Security-as-a-Service, by monitoring security parameters specified in SLAs and by providing the techniques to systematically manage SLA life-cycle. The project provided solutions for automatic negotiation and monitoring of SLAs between CSPs and SPECS platform based on security properties of cloud services. The work presented in this paper directly links with the outcomes of SPECS as MUSA extends these to multi-cloud setups.

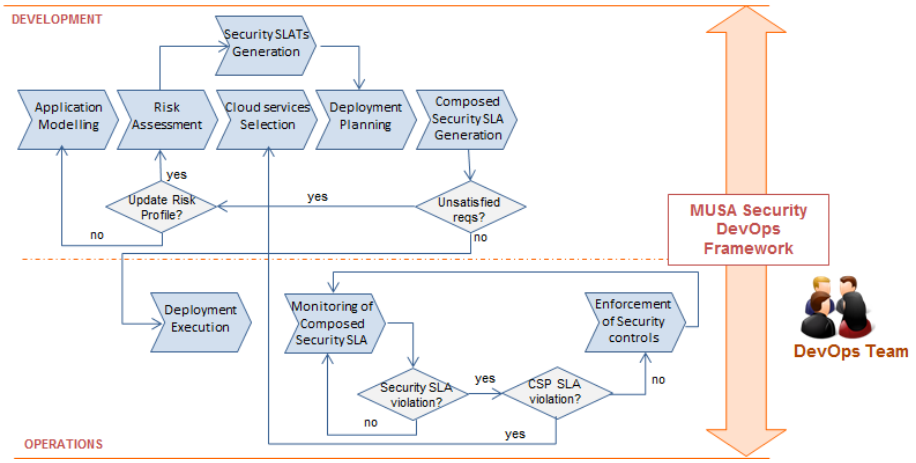

Fig. 2. Security DevOps workflow supported by the MUSA framework.

To the best of authors' knowledge, no previous work addresses security-by-design principles to multi-cloud application, identifying the security risks that multi-cloud applications are exposed to, expressing the security controls that can mitigate the risks in a machine-readable Security SLA, and finally, continuously monitoring the multi-cloud application composite security SLA.

4 The MUSA Security DevOps framework for multi-cloud

Our approach to continuous security assurance in multi-cloud relies in integrating a number of predictive and reactive security mechanisms in the application life-cycle. MUSA promotes the DevOps paradigm [29] since in our approach an multi-disciplinary team combining individuals from Development and Operations teams is the responsible for the multi-cloud application engineering and runtime administration. We name *DevOps Team* to such multi-disciplinary team that involves application architects, developers, security architects, business managers, service operators and system administrators. The expertise of the team members comes from diverse aspects of cloud system engineering and management. Even if the team works together in the whole life-cycle process, in each phase, from design to application operation, one of the roles in the DevOps Team may need to take the responsibility of the activity.

The MUSA DevOps approach is enabled by the MUSA framework which supports the MUSA workflow depicted in Fig.2.

The MUSA approach aims at aiding multi-disciplinary DevOps teams in multi-cloud application life-cycle. The workflow supported by the tools in the MUSA framework is iterative and involves the following activities:

1. *Application modelling*: The initial step in the multi-cloud application design is the specification of its Cloud Provider Independent Model (CPIM), a task

supported by the MUSA Modeller. The application CPIM is expressed in a MUSA extended CAMEL language and specifies the application cloud and security requirements in a level of abstraction independent from specific Cloud services and providers the application will use.

2. *Continuous Risk Assessment* that helps in the selection of the security controls and metrics that will be granted in the Security SLA and controlled at runtime. The activity follows a methodology similar to the one described in [30]. For each component the relevant threats are selected according to the component nature. The technical and business impact of the threat are evaluated, as well as risk minimization measures selected. These measures are defined as the desired countermeasures or controls required over the cloud services the components will use. The controls are expressed following the NIST standard Security Control Framework. The risk assessment is continuously updated with the feedback from the continuous monitoring of the controls behaviour at runtime.
3. *Cloud services selection*: In order to take most out of cloud services combination in terms of security, the DevOps Team is supported by the MUSA Decision Support Tool (DST) in the selection of cloud services that best match the security requirements of the application components. The best match is obtained by comparing the security controls offered by the cloud services under study (those previously categorized in the MUSA CSP Data Repository) with the security requirements of the individual components.
4. *Security SLA templates generation*: Once the most appropriate cloud service is selected for each of the components, the DevOps Team will use the MUSA SLA Generator to automatically create the Security SLA templates of the components. To this aim, the MUSA framework supports the verification of the feasibility of the components' Security SLA templates by checking whether the cloud service offerings selected in the previous step do offer such security requirements (in form of security controls). In case they do not, the MUSA security enforcement agents may be adopted to offer them.
5. *Deployment planning*: The Security SLA templates will be stored in the SLA Repository and will be retrieved by the MUSA Deployer so it can generate the multi-cloud deployment Implementation plan for the application. The Implementation plan specifies the application's software components to be installed and the cloud services to be provisioned, as well as their configuration details.
6. *Composed Security SLA generation*: In this step the DevOps Team is supported by the MUSA SLA Generator in the automatic generation of the final offered Security SLA of the overall multi-cloud application following the technique explained in section 2.2.
7. *Deployment execution*: The DevOps Team uses the MUSA Deployer to execute the previously created Implementation plan. The execution includes the provision and configuration of the needed cloud resources as well as the deployment of both the application components and the corresponding MUSA agents required in the plan.

8. *Continuous Monitoring of composed Security SLA*: Once the components are deployed and running, the MUSA Security Assurance platform starts monitoring the required metrics on the multi-cloud application components based on the final composed Security SLA. The continuous monitoring ensures the composed Security SLA holds and alarms are raised whenever incidents are detected.
9. *Enforcement of security controls*: As part of the possible reaction measures to detected incidents, the MUSA Security Assurance platform supports the dynamic enforcement of MUSA security enforcement agents. These agents are security mechanisms that can be activated to work with the component in case the detected problem is associated to the agent. Note that for the agents to work with the components, the components have to be prepared at design time.

In the MUSA workflow, Application modelling, Continuous Risk assessment and Cloud services selection follow an iterative loop that allows identifying whether there are any security requirements in the application that are not possible to be addressed with the security controls offered by the cloud services available. In this case, the DevOps Team should revisit the CPIM to include protection components or specify the use of MUSA security enforcement agents that offer such missing security controls (if available).

In the following we detail the MUSA solution for Security Assurance which includes the last two steps: Continuous monitoring and enforcement.

5 MUSA Security Assurance in Multi-cloud DevOps

Once all the components of the application have been appropriately deployed and the application is up and running, the runtime monitoring and operation of the multi-cloud application starts. The deployment scenarios may be diverse depending on application architecture, where some of the components may be deployed in cloud IaaS or may use PaaS or SaaS services. Therefore, the assurance solution in MUSA needs to be instantiated to select the right monitoring and security enforcement agents that fulfil the selected deployment environment's needs.

5.1 MUSA Security Assurance Workflow

In this section we zoom in the MUSA Security Assurance part of the MUSA workflow of Fig.2. There are two main activities included in the MUSA Security Assurance at operation that are explained in the following subsections.

Continuous Monitoring The objective of this activity is to monitor the security behaviour at operation of the multi-cloud application under test, in order to early react to possible incidents and violation of the Security SLA. The sub-activities are:

- Extraction of metrics and thresholds from Security SLAs: After retrieving the offered composite Security SLA registered in the MUSA Security Assurance Platform (in the Security SLA Repository of the platform), the platform will extract from it the security metrics that need to be monitored in the application components and in the cloud providers. The SLO thresholds that will apply for triggering the alerts and notifications are also learned from the Security SLAs.
- Configuration of monitoring agents: Make the needed arrangements and configurations for the MUSA monitoring agents to properly work and enable them to monitor the security metrics.
- Security metrics measurement and monitoring: Take the actual values of the metrics and store them in the Measurement Repository.
- Reporting and visualization of monitoring results: Show to the user the resulting values measured and the reports from the computation of the metrics.
- Notification of security incidents: The DevOps Team can subscribe to desired alerts and notifications. The envisaged notifications could be mainly of two types: Security SLA violations (when it is detected that a SLOs in the Security SLA is not reached) and alerts (when it is detected that a threshold level in the SLO is not reached, i.e. before any violation in the SLO occurs). The user will therefore need to set the threshold levels for the alerts.

Adaptation and reaction The goal of this activity is adapt to security incidents detected while monitoring in order to try to ensure the Security SLA of the multi-cloud application still holds. Therefore, the activity involves the decisions on and execution of the necessary reaction measures in case potential or actual Security SLA violations are detected. The reaction mechanisms in MUSA relies depend on the cause and type of the incident as well as whether it is an alert or a violation, i.e. the SLA is about to be violated or violation has effectively occurred already, respectively. The summary of the possible reaction measures that the DevOps Team can adopt is given below.

- *Activate a MUSA security enforcement agent*: The MUSA security enforcement agents are software artefacts provided by MUSA framework that implement one or more security controls in the multi-cloud application components that use them. In those cases that the component was prepared at design time to be able to use a MUSA enforcement agent, it is possible to enable such agent at runtime. The enforcement service in the MUSA Security Assurance Platform will identify the needed MUSA enforcement agent and activate it in the component in case the incident was detected in such component and corresponds to a security control implemented by the agent.
- *Re-deployment of multi-cloud application*: Whenever the cause of the security incident detected strives in a security control by a cloud service used by one or more of the components, the DevOps Team may need to replace the failing cloud service with some other cloud service offering similar functionality and security controls. The DevOps Team will therefore perform a new search

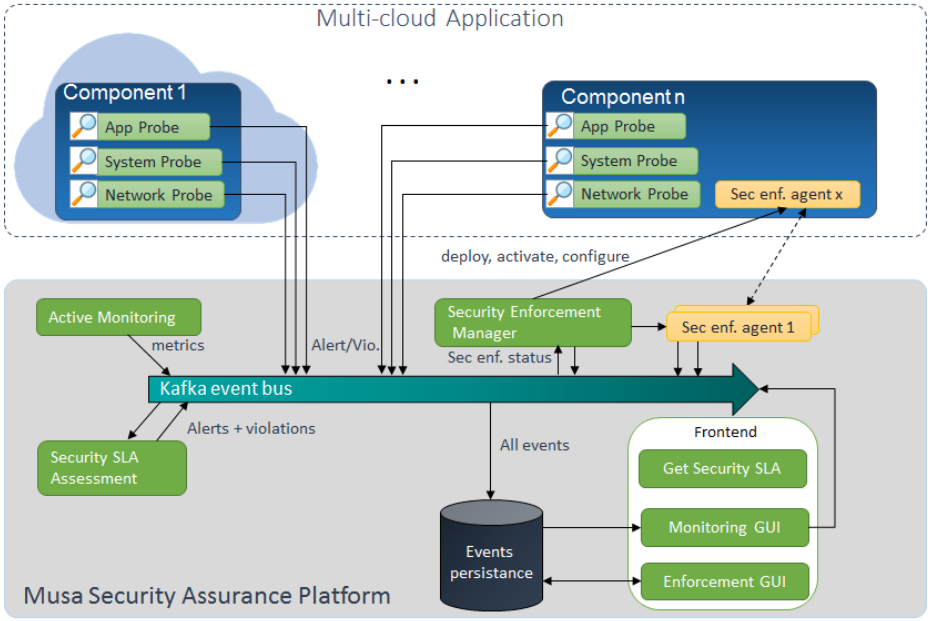

Fig. 3. MUSA Security Assurance Platform Architecture.

in the MUSA Decision Support Tool to find a replacement service and redeploy the components that were using the failing cloud service. Most likely the rest of the components may not be affected, but it is advisable the whole process starts from the beginning in order to make sure the new Cloud Service combination with the Cloud Service replacement still holds the multi-cloud application security requirements.

– *Re-design of multi-cloud application*: In case the cause of the security incident resides in an defective or poor security performance of one or more of the constituent components and not in the Cloud Services in use, the DevOps Team may need to update the multi-cloud application design and refine the security requirements or modify the components. This means that the DevOps Team will need to analyse the report of the causes and start the Design process again.

5.2 MUSA Security Assurance Platform architecture

The MUSA Security Assurance solution fits the operation phase of the MUSA framework. The MUSA Security Assurance Platform takes as input the Security SLA of the application to monitor the multi-cloud composite application SLA as well as the individual components' SLAs referred by it. From individual SLAs, the platform knows the right security metrics to monitor in single components

and from composite SLAs it is able to monitor the security of the overall application taking into account the communication exchanges between distributed components. In order to learn the list of monitoring agents deployed with each application component as well as their IP addresses, the MUSA Security Assurance Platform also requires the application deployment Implementation plan.

As shown in Fig.3, the MUSA Security Assurance Platform is composed of three main elements:

- The MUSA Monitoring agents responsible for collecting the security metrics specified in the Security SLA and relevant events to be analysed by the centralized MUSA Security Assurance platform.
- The MUSA Security enforcement agents that are deployed and/or activated in case of any security incident detection.
- The MUSA Security Assurance Platform that is deployed as Software-as-a-Service and allows collecting, displaying and managing all the security metrics and events from individual application components. The Platform gathers the measurements and events from the monitoring agents, checks the component Security SLAs and computes the composite metrics to check the global application Security SLA ("SLA checking" module). In case of an alert or a violation, the "security enforcement manager" is responsible for deploying, activating and configuring the local or remote security enforcement agents to mitigate the security risk. The communication with both monitoring and enforcement agents is done through the same KAFKA event bus.

In the following two subsections we present details of both the monitoring and enforcement agents.

MUSA Monitoring agents Three different types of monitoring agents can be deployed in the same virtual machine or container as the application component to compute security metrics by relying on different sources: Network, operating system or application. The security metrics that can be monitored thanks to the use of MUSA monitoring agents are those specified in the Security SLA of the application which relates to the individual components' Security SLAs.

- Network monitoring agent: This agent is responsible for analysing network traffic from different network interfaces of the virtual machine or container where the component is running. This agent facilitates network performance monitoring and operation troubleshooting through its real-time and historical data gathering. The agent correlates network events to detect performance, operational and security incidents thanks to its advanced rules engine.
- System monitoring agent: This agent monitors operating system resources to detect server performance degradation or performance bottlenecks early on. The agent relies on Linux *top* command to monitor Linux performance. The *top* command is available in many Linux/Unix-like operating systems

and is used to display all the running and active real-time processes, CPU usage, Memory usage, Swap Memory, Cache Size, Buffer Size, Process PID, User, among others.
- Application monitoring agent: It monitors information about the internal state of the target system, i.e., multi-cloud application component in which it is deployed. It notifies the MUSA security assurance platform about measurements of execution details and other internal conditions of the application component. The application monitoring agent is a Java library composed by two parts. The first part is an aspect to be weaved into the application code via pointcuts aimed at sending 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, providing a simple interface for sending log data to the MUSA Security Assurance Platform in a secure way.

MUSA Security enforcement agents The security enforcement agents offered in MUSA are security controls or mechanisms that can be easily integrated in multi-cloud application components and activated at runtime whenever needed to react to a violation in the Security SLA. The MUSA enforcement agents are built on top of existing open source solutions and the major innovation resides in having MUSA framework as single point of management for orchestrating multiple mechanisms in an homogenized way and from the same enforcement management dashboard. The enforcement agents can be deployed by the MUSA Deployer just as the individual components of the application. The MUSA Deployer interprets the application design model in MUSA extended CAMEL to learn the agents deployment configuration and execute it. Below we present two of the enforcement agents provided in the MUSA framework.

- *The high availability (HA) framework*: The HA framework is a collection of open-source software built around the Corosync/Pacemaker stack [31]. It provides high availability clustering mechanisms in multi-cloud for scalability, load balancing, automatic failover, automatic routing between services, and inter-component secure communications. Different deployment configurations of the HA framework are possible, together with the component in its docker container on IaaS nodes, or side-by-side with the component for both IaaS and PaaS systems.
- *The access control (AC) framework*: The AC framework ensures that only authorised parties can access and use the functionality offered by the components. Therefore, it offers access control in end-user-to-component communications and in component-to-component communications. The frameworks uses solutions external to MUSA to offer the authentication and authorisation functionality. The access control policies should be defined following the XACML [32] policy model. The Policy Enforcement Point (PEP) and Policy Decision Point (PDP) would be included in each component, which allows for taking the permission decisions locally and increase the performance.

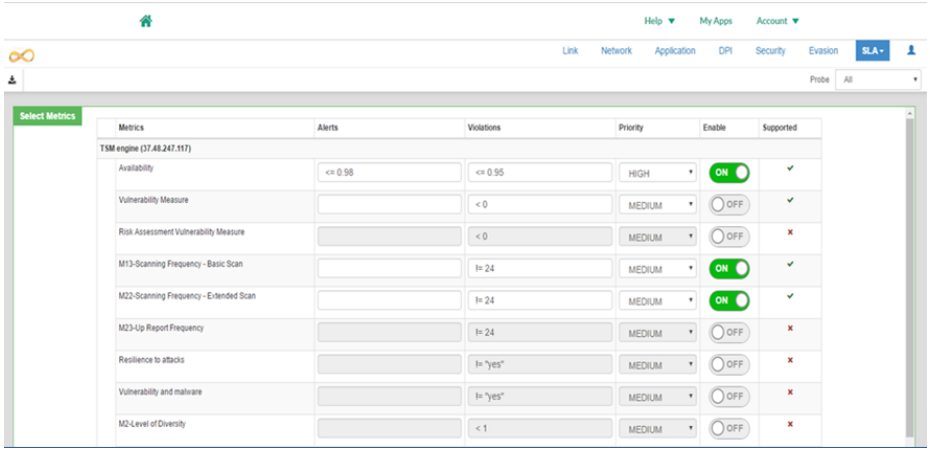

Fig. 4. Partial list of security metrics for the fleet module.

6 Validation

The MUSA framework and the SLA-based security assurance approach have been evaluated in the creation and operation of two real-world multi-cloud applications withing the MUSA proejct use cases:

- Smart mobility service by Tampere University of Technology, Finland. This is a multi-cloud application aims at supporting the energy efficient and sustainable multi-modal transport of Tampere citizens when commuting from home to work and vice versa. The application uses open data and services available in the Intelligent Transport Systems and Services (ITS) platform [33], where the Tampere City Council has a number of services exposed to allow companies and individual developers to develop, test and productize own traffic applications using public data. The services can be publicly accessed and include the public transport services APIs, other traffic related APIs, traffic data, etc. Multi-cloud applications that integrate IST Factory cloud-based services will be empowered with MUSA assurance tools for ensuring security of data storage and exchange at runtime. The focus of this case study is to ensure high availability of the application as well as confidentiality and integrity of citizens personal data.
- Flight scheduling application by Lufthansa Systems, Germany. This is a working prototype for a flight schedule planning application. The prototype is realized as a multi-layered, distributed web application and provides a scalable platform of self-contained and loosely coupled business components, each capable of running in a separate process and interacting by use of lightweight REST style communication protocols. The focus of this case study is on data integrity, confidentiality, localization and access control of different services within the application.

The DevOps Teams in both case studies followed the different steps of MUSA workflow to design the composite application, create its composite SLA and monitor it at runtime. In the process, individual SLA templates were created with the SLA Generator and Cloud service providers selected using the DST tool. After the application deployment planning and the computation of the composite application SLA, the application was deployed successfully using the MUSA Deployer tool and continuous monitoring and reaction was performed. In both use case scenarios, the MUSA monitoring agents described in section 5.2 were deployed to retrieve the security metrics specified in the applications' Security SLAs. For some of the components MUSA enforcement agents described in section 5.2 were deployed as well.

For instance, for the smart mobility service, one of the main application components is called *TSM Engine* which is a Web service responsible for retrieving diverse information (location, weather, paths, energy etc.) from the other components. After the risk analysis step, the Security SLA specified the following security controls (partial list), expressed according to the NIST Control Framework [13]:

- Denial of service protection (NIST SC-5)
- Vulnerability scanning (NIST RA-5)
- Authentication management (NIST IA-5)
- Session Authenticity (NIST SC-23)
- Information System Monitoring (NIST SI-4)
- etc.

And the following security metrics (partial list) where included in the Security SLA related to the security controls, as shown in Fig.4:

- Availability
- Vulnerability Measure
- Risk Assessment Vulnerability Measure
- M13-Scanning Frequency - Basic Scan
- M22-Scanning Frequency - Extended Scan
- M23-Up Report Frequency
- Resilience to attacks
- etc.

The continuous monitoring of the security metrics in the Security SLA allowed detecting potential malicious activities based on a set of detection rules denoting several kinds of attack signatures. For example, to evaluate the efficiency in availability, we stopped the Tomcat server where the TSM engine was running which caused the application availability rate (SLO more than 99%) stated in the Security SLA was not respected. The incident was detected by the MUSA monitoring agent and notified to the MUSA Security Assurance Platform that raised immediately a violation alarm to the DevOps Team together with the recommendation to restart the Tomcat server and to ensure server redundancy. The DevOps Team followed the advice and used the *High availability framework*

MUSA enforcement agent to ensure server redundancy, which helped to recover from the incident. Additional monitoring and reaction strategies provided by the solution for incidents on identity thefts, unauthorized access control threats, etc. were evaluated successfully.

7 Conclusion and Future work

Multi-cloud applications pose a great challenge to security assurance due to they have to tackle the security of the individual components as well as the overall application security, which in turn depends on the security controls provided by the cloud services in use. Despite the cloud service providers offer their own security controls, the multi-cloud application has to ensure integrated security across the whole composition.

The MUSA DevOps framework has been conceived and prototype created to address such challenge. It supports the security-aware design and operation of multi-cloud applications and integrates security-by-design mechanisms with continuous security assurance. The later is offered in form of a Software-as-a-Service solution named *MUSA Security Assurance Platform*.

The MUSA Security SLA-based assurance advances over the state of the art in security-aware cloud SLAs, which foster clarity and transparency in cloud service provisioning. The application composite Security SLA can be computed and expressed in machine-readable format, relying on standard security control families like those of NIST and Cloud Security Alliance.

The MUSA Security Assurance platform enables cloud transparency by being able to monitor security metrics in the Security SLA, and by keeping informed multi-cloud application consumers on the real-time behaviour of both the components and the cloud services underneath. Non-compliance with respect to security level objectives in the Security SLAs of the used CSPs and the application are early detected and reaction measures activated for prompt mitigation of the risks. As a result, the presented approach enables multi-cloud applications be secure and self-adaptive.

The initial evaluation of MUSA security assurance approach in two real case studies showed that the proposed methods and tools reduce the security flaws in the application implementation and are effective in ensuring the multi-cloud application complies with its composite Security SLA. Further evaluation rounds of the MUSA framework are planned in the next months within the two case studies presented above in order to assess the effectiveness and usability of the framework tools, particularly in the support to an integrated and multi-disciplinary DevOps approach to enhance security in multi-cloud applications and rise consumers' trust in cloud-based environments.

Acknowledgment

The MUSA project leading to this paper has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement

No 644429. We would also like to acknowledge all the members of the MUSA Consortium for their valuable help.

References

1. Rightscale: Cloud computing trends: 2017 state of the cloud survey (2017) Available at: http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2017-state-cloud-survey.
2. Deloitte: Measuring the economic impact of cloud computing in europe, smart number: 2014/0031 (2017) Available at: https://ec.europa.eu/digital-single-market/en/news/measuring-economic-impact-cloud-computing-europe.
3. Casola, V., De Benedictis, A., Modic, J., Rak, M., Villano, U.: Per-service security sla: a new model for security management in clouds. In: Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2016 IEEE 25th International Conference on, IEEE (2016) 83–88
4. Blasi, L., Brataas, G., Boniface, M., Butler, J., DAndria, F., Drescher, M., Jimenez, R., Krogmann, K., Kousiouris, G., Koller, B., Landi, G., Matera, F., Menychtas, A., Oberle, K., Phillips, S., Rea, L., Romano, P., Symonds, M., Ziegler, W.: Cloud computing service level agreements – exploitation of research results (06 2013) Available at: http://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?doc_id=2496.
5. SPECS project: Secure Provisioning of Cloud Services based on SLA management. FP7-ICT-2013.1.5 (2013-2016. Available at: www.specs-project.eu/)
6. SLALOM project: Service Level Agreement - Legal and Open Model (2015-2016. Available at: www.slalom-project.eu/)
7. SLA-READY project: Making Cloud SLAs readily usable in the EU private sector. (2015-2016. Available at: www.sla-ready.eu/)
8. Rios, E., Iturbe, E., Orue-Echevarria, L., Rak, M., Casola, V., et al.: Towards self-protective multi-cloud applications: Musa–a holistic framework to support the security-intelligent lifecycle management of multi-cloud applications. (2015)
9. MUSA project: Multi-cloud Secure Applications (2015-2017) Available at: http://www.musa-project.eu.
10. Forum, G.I.C.T.: Use cases and functional requirements for inter-cloud computing. In: Global Inter-Cloud Technology Forum, GICTF White Paper. (2010)
11. ISO/IEC: Iso/iec 20000-1:2011 information technology – service management – part 1: Service management system requirements (2011) Available at: https://www.iso.org/standard/51986.html.
12. ETSI: Interoperability and security in cloud computing, etsi sr 003 391 v2.0.0 (2015) Available at: http://csc.etsi.org/resources/WP3-Report/STF_486_WP3_Report-v2.0.0.pdf.
13. National Institute of Standards and Technology (NIST): Security and privacy controls for federal information systems and organizations. **800-53** (apr 2013) 1–460
14. Alliance, C.S.: Cloud security alliance, cloud controls matrix v3.0.1 (2016) https://cloudsecurityalliance.org/research/ccm/.
15. ISO/IEC: Iso/iec 27017:2015 information technology – security techniques – code of practice for information security controls based on iso/iec 27002 for cloud services (2015) Available at: https://www.iso.org/standard/43757.html.
16. Casola, V., De Benedictis, A., Rak, M., Modic, J., Erascu, M.: Automatically enforcing security slas in the cloud. IEEE Transactions on Services Computing (2016)

17. Andreieux, A.: Web services agreement specification (2007) Available at: https://www.ogf.org/documents/GFD.107.pdf.
18. Casola, V., Benedictis, A.D., Rak, M., Villano, U.: A security metric catalogue for cloud applications. In: Complex, Intelligent, and Software Intensive Systems - Proceedings of the 11th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2017), Torino, Italy, July 10-12, 2017. (2017) 854–863
19. Liu, H., Bu, F., Cai, H.: Sla-based service composition model with semantic support. In: Services Computing Conference (APSCC), 2012 IEEE Asia-Pacific, IEEE (2012) 374–379
20. Zappatore, M., Longo, A., Bochicchio, M.A.: SLA composition in service networks. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing - SAC '15, New York, New York, USA, ACM Press (2015) 1219–1224 Available at: http://dl.acm.org/citation.cfm?doid=2695664.2699490.
21. Rak, M.: Security assurance of (multi-) cloud application with security sla composition. In: International Conference on Green, Pervasive, and Cloud Computing, Springer (2017) 786–799
22. Fatema, K., Emeakaroha, V.C., Healy, P.D., Morrison, J.P., Lynn, T.: A survey of Cloud monitoring tools: Taxonomy, capabilities and objectives. Journal of Parallel and Distributed Computing **74**(10) (oct 2014) 2918–2933
23. Naser, A., Zolkipli, M.F., Anwar, S., Al-Hawawreh, M.S.: Present Status and Challenges in Cloud Monitoring Framework: A Survey. In: 2016 European Intelligence and Security Informatics Conference (EISIC), IEEE (aug 2016) 201–201
24. Rak, M., Venticinque, S., Mhr, T., Echevarria, G., Esnal, G.: Cloud Application Monitoring: The mOSAIC Approach. In: 2011 IEEE Third International Conference on Cloud Computing Technology and Science, IEEE (nov 2011) 758–763
25. Zeginis, C., Kritikos, K., Garefalakis, P., Konsolaki, K., Magoutis, K., Plexousakis, D.: Towards Cross-Layer Monitoring of Multi-Cloud Service-Based Applications. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Volume 8135 LNCS. (2013) 188–195
26. Brogi, A., Ibrahim, A., Soldani, J., Carrasco, J., Cubo, J., Pimentel, E., D'Andria, F.: Seaclouds: a european project on seamless management of multi-cloud applications. ACM SIGSOFT Software Engineering Notes **39**(1) (2014) 1–4
27. SINTEF: Model-based provisioning and deployment of cloud-based systems (2013) Available at: http://cloudml.org.
28. Columbus, L.: Computerworld's 2015 forecast predicts security, cloud computing and analytics will lead it spending. Online at: http://www.forbes.com/sites/louiscolumbus/2014/11/26/computerworlds-2015-forecast-predicts-security-cloud-computing-and-analytics-will-lead-it-spending/ (2014)
29. Gartner, I.: Gartner it glossary devops. Gartner IT Glossary (2017) Available at: http://www.gartner.com/it-glossary/devops.
30. Afolaranmi, S.O., Moctezuma, L.E.G., Rak, M., Casola, V., Rios, E., Lastra, J.L.M.: Methodology to obtain the security controls in multi-cloud applications. In: Proceedings of the 6th International Conference on Cloud Computing and Services Science - Volume 1: CLOSER,, INSTICC, ScitePress (2016) 327–332
31. Openstack: Pacemaker cluster stack (2015) https://docs.openstack.org/ha-guide/controller-ha-pacemaker.html.

32. OASIS: extensible access control markup language (xacml) version 3.0 (2013) Available at: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.
33. of Technology, T.U.: Intelligent transport systems and services (its) factory developer wiki (2014) Available at: http://wiki.itsfactory.fi/index.php/ITS_Factory_Developer_Wiki.