SLA-driven Monitoring of Multi-Cloud Application Components using the MUSA framework

Erkuden Rios[1], Wissam Mallouli[2], Massimiliano Rak[3], Valentina Casola[3], Antonio M. Ortiz[2]
[1] ICT Division, Fundacion Tecnalia Research & Innovation. Derio, Spain
Email: erkuden.rios@tecnalia.com
[2] Montimage R&D. Paris, France
Email: {wissam.mallouli, antonio.ortiz}@montimage.com
[3] University of Naples, Naples, Italy
Email: massimiliano.rak@unina2.it, casolav@unina.it

***Abstract*—The applications that rely on the combined use of multiple independent clouds pose a challenge to the control of their security. The difficulty of this task resides on the lack of insight on the cloud providers security measures, plus the need to simultaneously monitor the behaviour of multiple individual components deployed in different clouds. This paper presents the SLA-driven monitoring of multi-cloud application security compliance. In this approach, the application security levels, controls and metrics are specified at design time in the Service Level Agreement (SLA) creation process and continuously monitored at runtime once the application components are deployed over the multi-cloud. The security monitoring is based on the Montimage Monitoring Tool (MMT), that combines Deep Packet Inspection (DPI) and data mining techniques to collect and analyse measurements at both network and application component levels for a holistic assurance.**

***Keywords*—** Security SLA generation, multi-cloud application, monitoring, SLA checking.

I. Introduction

The applications that are able to exploit at the same time multiple independent clouds will most likely master the cloud market in the sense that they will be the ones getting the right resources and features when needed and at the minimum cost. These multi-cloud applications however pose a challenge to the control of their security. The difficulty of the task resides on the lack of insight on the cloud providers' security measures plus the need to simultaneously monitor the behaviour of multiple individual components deployed in diverse clouds.

The MUSA EU Horizon 2020 project[1] is currently developing a framework that includes methods and tools to tackle this task. In fact, the MUSA framework aims at supporting the security-intelligent life cycle management of distributed applications over heterogeneous cloud resources.

In this work we focus on the methods and tools included in the MUSA framework that support the SLA-driven monitoring of the multi-cloud application components, particularly, the focus is on the MUSA support to the generation of the security SLAs of multi-cloud applications, and the security SLA monitoring and notification at runtime.

The remainder of this paper is structured as follows: The next section II describes the state-of-the-art together with the major security challenges faced by applications or services working in multi-cloud environments. Then, section III introduces the MUSA framework in order to contextualise the methods and tools proposed herein and focuses on the security SLA modelling and generation. The Section IV describes in detail the MUSA Security Assurance Platform that is the core module of the MUSA framework in charge of runtime security monitoring and enforcement. In Section V we explain the preliminary results of our work that is being applied to a smart mobility service (based on multi-cloud) for the Tampere citizens. Finally, Section VI concludes the paper and explains future work.

II. multi-cloud Security Challenges

According to the taxonomy proposed by [1] and [2], 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. At state of art, few concrete multi-cloud solutions exist, addressed in research projects[2] like OPTIMIS, mOSAIC, MODAClouds, PaaSAge, Cloud4SOA [3], [4]. 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: [1], [5] and [6].

In the literature, several research works judge that relying on multi-cloud solutions can improve security. Others believe that, on the contrary, this will bring new security risks and vulnerabilities. For instance, on one hand, the authors of [7] and [8] offers simple surveys of solutions that try to improve the security using multi-cloud techniques, as an example [9] and [10] proposes techniques to distribute a file over multiple providers or untrusted networks, granting higher confidentiality and data integrity. On the other hand, [11] and [12] face the security in multi-cloud application from different perspectives: they analyse different multi-cloud solutions and try to make a security assessment of the overall application behaviour. According to such vision, multi-cloud is open to new security

[1] http://www.musa-project.eu/

[2] OPTIMIS: http://www.optimis-project.eu/, mOSAIC: http://www.mosaic-cloud.eu/, MODAClouds: http://www.modaclouds.eu/, PaaSAge: http://www.paasage.eu/, Cloud4SOA: http://www.cloud4soa.com/

threats that decrease the global security level. To the best of the authors' knowledge there are no concrete techniques that try to address the issue of developing multi-cloud applications trying to take into account the user security requirements from the early development stages.

SLA monitoring and reporting is a classical activity that is generally performed by third party monitoring tool providers to assess the contract terms [13], [14]. The monitoring of security SLAs has been addressed by several research works like in [15] where the authors define a monitoring architecture integrating different security-related monitoring tools to collect measurements of specific metrics associated with the set of security Service Level Objectives (SLOs) that have been specified in the Security SLA. In our paper, we will reuse these monitoring techniques and extend them to runtime monitoring of SLAs during application operation by using deep packet inspection (DPI) technology implemented in Montimage Monitoring Tool [16].

III. Introduction to the MUSA framework

A. Security SLAs for multi-cloud based Applications

The MUSA framework aims at offering a solution to develop multi-cloud applications adopting a security-by-design approach: application development tries to take into account security problems from the very early development stages. In MUSA, a multi-cloud application, as shown in Fig. 1, is modelled in terms of a set of components, which can be executed independently and that interact with each other during the multi-cloud application execution. Each component uses SaaS (Software-as-a-Service) cloud services and/or is hosted by IaaS (Infrastructure-as-a-Service) cloud services.

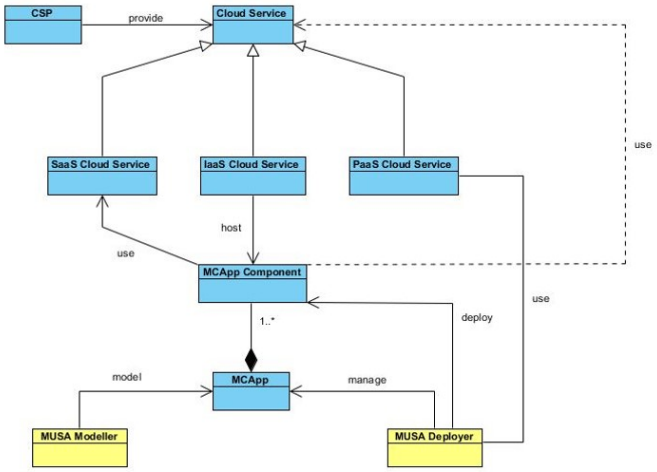

Fig. 1. The multi-cloud application model

In order to address the security-by-design approach, MUSA adopts the concept of security Service Level Agreement (security SLAs): A Service Level Agreement is a contract among service provider and customer that states the quality level of the services provided. A security SLA contains Service Level Objectives (SLOs), i.e., grants measurable by the customers, related to the security of the provided services.

The MUSA Development Framework supports the generation of the needed security SLAs, each corresponding to an individual multi-cloud application component. Such security SLAs are then stored in a repository so that the MUSA Security Assurance Platform can verify that the promised SLOs are respected during the multi-cloud application lifetime. In this paper, we will just summarise the process of security SLA generation and the demonstration of its feasibility, in order to concentrate on the monitoring aspects, demonstrating that, once the application is in execution, it is possible to monitor its security according to the security SLAs.

In the following section we briefly summarise the main concepts related to security SLAs and their generation process. In this paper we do not focus on the composition problem (i.e., on what is effectively granted to the full multi-cloud application), but only on the correct identification of the security requirement of each component. Such security SLAs can be used to drive the monitoring process that, at runtime, enables to verify that components are effectively respecting the granted security levels and notify to application users possible issues.

B. Security SLA Model

In order to represent security in SLAs we adopted the SLA format proposed in SPECS ([17], [18], based on the WS-agreement standard [3]).

As illustrated in Fig. 2, the proposed *wsag* extensions represent security into an SLA using a few simple concepts:

- **Security capabilities**: the set of security controls[4] that a security mechanism is able to enforce over the target service.
- **Security metrics**: the measurement standards adopted to evaluate the security levels of the offered services.
- **SLOs**: the conditions, expressed over security metrics, representing the security levels that must be respected according to the SLA.

It is worth noting that **security capabilities** are a *declarative* section of the SLA, which cannot be directly measured, but describe the service security according to common best practice: as a set of standard security controls adopted in service implementation in order to address security issues.

Instead, **security metrics** and **SLOs** are the measurable part of the SLA, which can be monitored and verified by both parties of the agreement.

SLOs and metrics are *associated* to the declared **security capabilities** in order to offer a quantitative measure of the declared security controls, through the ws-agreement concept of properties (which we have not reported in the figure for simplicity).

C. SLA creation process using the MUSA Framework

The MUSA Framework proposes a DevOps approach for the design, development, deployment and operation manage-

[3]https://www.ogf.org/documents/GFD.107.pdf

[4]NIST standard NIST800$_5$3r4 : $http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf$

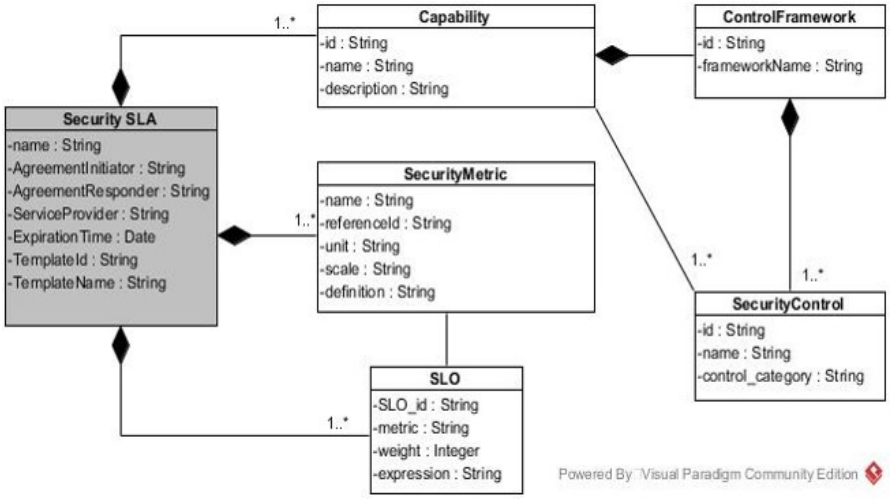

Fig. 2. The security SLA model

ment of the multi-cloud applications. In this sense, the tools proposed allow for a seamless support of both the development and operation tasks, including runtime monitoring and control. The specification of the security-aware model of the application and the creation of its SLA, being them design-time tasks, get seamlessly integrated with security assurance at operation time. In MUSA, the process of multi-cloud application SLA creation is supported by two tools named SLA Generator and the Decision Support Tool (DST). Indeed, what it is created is the security SLA (the SLA extended with security SLOs and controls) of the composite application. The process starts by interpreting the architectural information contained in the Cloud Provider Independent Model (CPIM) of the multi-cloud application, and generating the Security SLA template (without specific CSP information) with the desired cloud service type assigned to each of the multi-cloud application components. Then, the SLA template undergoes a refinement process where the required security controls and metrics are specified. In order to do so, an initial risk analysis needs to be done with the help of the DST. In this risk analysis, the organisation assets are defined together with their importance and the risks over them are analysed. The result of the analysis is the establishment of the priorities of the wanted security controls in the multi-cloud application.

With this analysis the DevOps team details the initially desired security controls and metrics in the security SLA template. In order to complete the security SLA with the proper controls and metrics required over specific selected CS offerings, an iterative process starts until all requirements of the multi-cloud application are expressed in the security SLA. In this loop, the DST is consulted in order to learn on the best candidate Cloud Service combinations that can offer the specified security controls to fulfil the multi-cloud application security requirements. The candidate combinations are ranked (according to the risk profile and the requirements satisfaction rate) and presented to the DevOps team who make the final decision on which CS should be selected. Once (some of) the CSs are selected, the information on actual controls and metrics offered by the selected CS needs to be detailed in the security SLA. The indication of the needed monitoring agents (per security property and per multi-cloud application component) is also optionally added. The Security SLA is increasingly refined by checking the available security controls of the CS and indicating the desired ones. When all the security controls and metrics for individual components are specified, the composite SLA is created and a final check is done on whether all the multi-cloud application requirements can still be satisfied by the selected CS combination. The composite security SLA refinement process will continue until all the requirements specified in the security SLA are fulfilled by the selected CS combination.

IV. MUSA Security Assurance Platform

A. Runtime Security Assurance Process

In the process for the runtime security assurance of the multi-cloud application, the DevOps will use the multi-cloud application monitoring tools to control both its performance and security behaviour. MUSA will offer them the MUSA Security Assurance Platform (SaaS) for the assurance of the security aspects. The DevOps team will first use the subscription service of the notification in MUSA SaaS to select which aspects they would like to receive notifications from. Thus, the notification service can forward/show notifications to subscribers according to the subscription criteria such as the alert/notification types, thresholds and reaction measures. Then, the MUSA Security Assurance Core will get the multi-cloud applications security SLA in force from the security SLA repository, and extract from it the security metrics that should collect and control in each of the multi-cloud application components and the overall multi-cloud application. The monitoring service is continuously checking those security properties (both properties of the components and of the

overall application) and storing the measurements (metrics values) in the Measurement Repository.

The monitoring service is also in charge of computing the needed composite security metrics. The Core will make all the necessary computations to evaluate whether all the clauses in the security SLA are being satisfied and if not, an appropriate notification order will be sent to the Notification service (alerts in case the SLA is in risk of being violated or violations in the case the SLA has indeed been violated) which then relays the notification to the subscribers.

Depending on the type of security flaw detected, the MUSA Security Assurance platform will offer different reactive measures. In some cases, the potential Security SLA violation may be corrected by means of the application of some enforcement mechanism in the multi-cloud application components. In such cases, the Enforcement service will participate by activating the needed enforcement agents in the components.

B. Monitoring Tools

The monitoring of security SLAs in MUSA relies on the use of multiple solutions(either developed ad-hoc or already available as open-source or commercial products) to retrieve necessary metrics and indicators to check their validity. These solutions include the tools proposed in the SPECS project [15] and extend them by a runtime monitoring solution composed by a set of :

- monitoring agents, deployed in different cloud components to continuously capture and analyse network communication as well as system status (CPU and memory usage) and application logs.
- monitoring libraries, part of the MUSA Security Assurance central platform, that allow to combine data captured from different agents and compute security-related metrics to check the conformity of SLAs. Furthermore, these libraries are able to trigger security alerts/violations based on the event rules generated from the parsing of SLOs.

These monitoring agents and libraries are provided by the Montimage Monitoring Tool (MMT). This tool is composed of three complementary, but independent, modules as shown in Fig. 3.

- MMT-Extract is the core packet processing module. It is a library that analyses network traffic using Deep Packet/Flow Inspection (DPI/DFI) techniques in order to identify network and application based events by analyzing: protocols' fields values; network and application QoS parameters; and, Key Performance Indicators (KPI). In a similar way, it also allows analysing any structured information generated by applications (e.g., traces, logged messages). MMT-Extract incorporates a plugin architecture for the addition of new protocols or messages, and a public API for integration into third party probes.
- MMT-Correlation is a security analysis engine based on MMT security properties. MMT-Correlation analyses and correlates network and application events to detect operational and security incidents. For each occurrence of a security property, MMT-Correlation allows detecting whether it was respected or violated. A set of security properties for SLAs checking has been specified within this tool to analyse their respect by the multi-cloud based applications.
- MMT also provides two other main functionalities. First, a user interface to allow operators to administrate different MMT probes. The second one is to collect and aggregate information gathered by different probes including security incidents to present them via a graphical user interface.

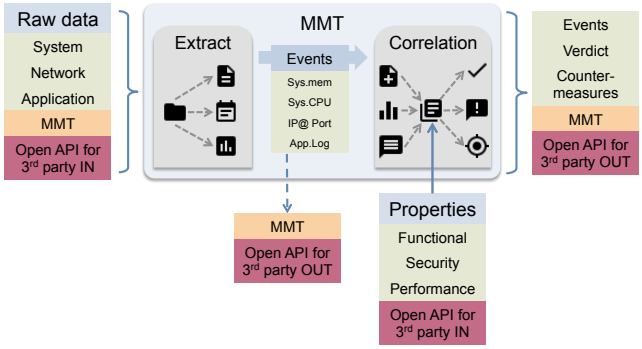

Fig. 3. MMT global architecture.

In the context of MUSA, MMT tool has been adapted to be deployed in multi-cloud environments. MMT is included in the software image of the virtual machine and it is thus automatically initiated when instantiating each virtual machine running an application component with no further configuration needed.

The MMT modules described above are deployed in the context of MUSA to detect potential SLAs violations. In order to reach this objective, the following 3 steps and extensions are performed.

Step1: MMT is deployed as part of the MUSA assurance platform. It allows to parse the required application components SLAs in order to extract the necessary metrics that need to be monitored by the MMT probes (MMT-Extract).

Step2: The MMT-Extract modules are deployed in different cloud provides and constitute the MMT-agents that are responsible to collect data (from the network, system or application levels) and send them to centralised MUSA assurance platform.

Step3: MMT-Correlation is deployed in the MUSA assurance platform. It intercepts the extracted metrics provided by different MMT-Extract modules and correlates them in order to compare them with the desired SLOs specified as part of the SLAs. In case of a violation, a notification is sent to the DevOps team.

V. Preliminary results - Tampere Smart Mobility Application

Tampere is the second largest town in Finland. Tamperes transportation structure is multi-modal, and the city has a

population of nearly half a million. The smart mobility multi-cloud application is called TSM (stands for Tampere Smart Mobility) and provides efficient and optimal transportation information to commuters by taking into consideration road, traffic and weather conditions, etc. The TSM application utilizes resources, services and information from the FMI (Finnish Meteorological Institute), Google directions and the Tampere Intelligent Transport System and Services (ITS) platform, in order to provide support for an energy efficient, optimal and sustainable multi-modal transportation for Tampere citizens. In addition to journey options, the mobility profiles of every user are also stored by the TSM application. All user activities related to time, money, travel distance and energy consumption shall be stored by the application. The TSM application is then capable of learning the activity patterns carried out by every user and by so doing, shall be able to provide recommendations based on the users frequent activities. The interaction of the TSM with other services using the commuters personal sensitive data like user mobility profile requires that privacy, security and protection of user data is implemented. To this end, adequate security controls such as the security of communication channels and data storage will be provided by the MUSA framework and integrated into the TSM application for the interaction between its components and also other services. Authenticated access shall also be implemented for applications that need to make use of personal users data.

A. SLA Production

In order to produce the SLA for each TSM application component, a risk analysis is performed to identify the potential threats that can cause damage, harm or loss in the studied application. CerICT build an on-line tool that allow to identify risks coming from these threats and propose a set of security capabilities that are linked to a set of security metrics to be monitored by the monitoring tool. The tool is available as a draft version on 37.48.247.125/TESI-0.0.1-SNAPSHOT/. The generated SLA is an XML format following the NIST standard. It contains for each application component, the list of threats, security controls and metrics with SLOs (security level objectives).

B. Monitoring mechanisms and integration

In the context of TSM case study, a set of security metrics has been defined. In Tab. I, eleven of them are presented.

For each metric, a monitoring methodology has been conceived. For instance, for the first metric related to the number of detected attacks by an IDS, MMT is deployed as an IDS in each application component. The MMT agent captures the network traffic and detects known attacks (specified as a set of security detection rules). Each detected attack is sent to the MUSA assurance platform to be notified to the application DevOps team. Other metrics, like service/data availability or application response time or access control and enforcement allow us to detect performance incidents that could be caused by an attack (generally deny of service attack). These metrics are also based on the analysis of network traffic. If the network communication is encrypted, the application log file is analysed. A connector has been implemented and integrated into the running Java application components to notify application internal events to MMT agent that computes required security metrics.

The combination of data extracted from different sources (network, application, system) allowed to compute the required metrics and check if they respect the desired service level objectives (SLOs) in the MUSA assurance central platform.

C. First results and future work

The analysis of the defined metrics for Tampere Smart Mobility (TSM) use case allowed to classify them into two big sets:

- Monitorable metrics: This is the case of most of the metrics where we can use an agent running in the same cloud infrastructure to collect data and retrieve the necessary information. This can be realised by relying of different monitoring tools. Some of them are lan
- Non monitorable metrics: This is the case of geolocation of application data. Indeed, no monitoring tool exists yet to allow the identification of data location. Besides, if no application internal data is logged, it becomes difficult and even impossible to know the type of encryption algorithms that are used to share data.

At this analysis level, the SLAs monitoring has been possible for individual SLAs, each SLA providing the service level agreement for a specific application component. The analysis of composite SLAs becomes a new challenge in terms of SLA generation and collaborative monitoring. This challenge will be addressed in the future work of this research work. Please refer to our work in [19] to learn on examples of security controls used and how the they are derived from the application risk analysis.

VI. Conclusion

In this paper, we presented the solution from the MUSA framework that focuses on i) the generation of security SLAs for a multi-cloud application based on a risk assessment of application components, and ii) the monitoring of these SLAs by relying on a set of monitoring tools, especially on Montimage Monitoring Tool.

The SLA generation is supported by a Web-based application used by the DevOps team to collect the security requirements of the multi-cloud application and to create the machine-readable SLA. The SLA generation application proposes, based on a risk analysis of individual application components, a set of security controls that can be monitored at runtime. The basic idea behind this monitoring is to check that the security requirements stated in the security SLAs are respected and raise an alert if not. In this case, the DevOps team is notified about the SLA violation in order they can take the necessary countermeasures to ensure security (e.g., by enforcing new security mechanisms provided by MUSA framework).

The SLA monitoring solution in MUSA is still under development and test and preliminary results allowed us to

Title	Description
Number of Detected Attacks	Number of detected attacks using a Intrusion Detection System (IDS).
Infrastructure Location	Defining the geo-location of data and infrastructure for governance and compliance purposes
Data availability	Percentage of time in which data access is available to data owners.
Service availability	Percentage of time in which service access is available to users.
Application response time	The average time (in milliseconds) to answer a specific request (can be categorised by request).
Service disruption	Deviation from the normal use regarding different metrics (response time, number of requests, number of connected users etc.) at different day periods.
Resilience to attacks	The application should be attack-tolerant.
Access control and enforcement	The application reports the number of valid access attempts, failed access attempts, access retries and also frequency of password change attempts.
Data encryption	The cryptographic mechanism checks if the information being sent or the data being stored in the cloud storage is encrypted and not in plain text. Also the percentage of encrypted stored data in the cloud infrastructure.
SQL injection	By monitoring the queries, it is possible to identify SQL injection attempts.
Database activity monitoring	Monitors the activity of privileged users (superusers) in databases and recognises abnormal behaviour.

TABLE I
SHORT LIST OF SECURITY METRICS FOR TSM CASE STUDY

check component level SLAs. The composite SLA generation and monitoring bring new challenges that will be targeted in future works.

ACKNOWLEDGMENT

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

REFERENCES

[1] Nikolay Grozev* and R. Buyya, "Inter-Cloud architectures and application brokering: taxonomy and survey," *Software - Practice and Experience*, vol. 44, no. 3, pp. 369—-390, 2012.

[2] Global Inter-cloud Technology Forum, "Use Cases and Functional Requirements for Inter-Cloud Computing," Tech. Rep., 2010.

[3] D. Petcu, "Multi-cloud: Expectations and current approaches," in *Proceedings of the 2013 International Workshop on Multi-cloud Applications and Federated Clouds*, ser. MultiCloud '13. New York, NY, USA: ACM, 2013, pp. 1–6. [Online]. Available: http://doi.acm.org/10.1145/2462326.2462328

[4] N. Ferry, A. Rossini, F. Chauvel, B. Morin, and A. Solberg, "Towards Model-Driven Provisioning, Deployment, Monitoring, and Adaptation of Multi-cloud Systems," *2013 IEEE Sixth International Conference on Cloud Computing*, pp. 887–894, 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6740239

[5] G. Baryannis, P. Garefalakis, K. Kritikos, K. Magoutis, A. Papaioannou, D. Plexousakis, and C. Zeginis, "Lifecycle Management of Service-based Applications on Multi-clouds: A Research Roadmap," in *Proceedings of the 2013 International Workshop on Multi-cloud Applications and Federated Clouds*, ser. MultiCloud '13. New York, NY, USA: ACM, 2013, pp. 13–20. [Online]. Available: http://doi.acm.org/10.1145/2462326.2462331 http://dl.acm.org/citation.cfm?id=2462331

[6] C. Zeginis, K. Kritikos, P. Garefalakis, K. Konsolaki, K. Magoutis, and D. Plexousakis, "Towards Cross-Layer Monitoring of Multi-Cloud Service-Based Applications," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8135 LNCS, pp. 188–195, 2013.

[7] M. Alzain, B. Soh, and E. Pardede, "TMR-MCDB: Enhancing Security in a Multi-cloud Model through Improvement of Service Dependability," pp. 133–144, jun 2014. [Online]. Available: http://www.iaesjournal.com/online/index.php/IJ-CLOSER/article/view/6294

[8] D. Bernstein and D. Vij, "Intercloud security considerations," *Proceedings - 2nd IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2010*, pp. 537–544, 2010.

[9] Z. Yan, H. Hongxin, A. Gail-Joon, and Y. Mengyang, "Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage," *IEEE Transactions on Parallel and Distributed Systems,*, vol. 23, no. 12, pp. 2231–2244, 2012.

[10] P. F. Oliveira, L. Lima, T. T. V. Vinhoza, J. Barros, and M. Medard, "Trusted Storage over Untrusted Networks," *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–5, 2010.

[11] J.-M. Bohli, N. Gruschka, M. Jensen, L. L. Iacono, and N. Marnau, "Security and Privacy-Enhancing Multicloud Architectures," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 4, pp. 212–224, 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6487347

[12] M. Singhal, S. Chandrasekhar, T. Ge, R. Sandhu, R. Krishnan, G.-j. Ahn, and E. Bertino, "Collaboration in Multicloud Computing Environments: Framework and Security Issues," *IEEE Computer*, pp. 76–84, 2013.

[13] N. Goel, N. V. N. Kumar, and R. K. Shyamasundar, "SLA monitor: A system for dynamic monitoring of adaptive web services," in *9th IEEE European Conference on Web Services, ECOWS 2011, Lugano, Switzerland, September 14-16, 2011*, G. Zavattaro, U. Schreier, and C. Pautasso, Eds. IEEE, 2011, pp. 109–116. [Online]. Available: http://dx.doi.org/10.1109/ECOWS.2011.22

[14] D. Petcu and C. Craciun, "Towards a security sla-based cloud monitoring service," in *CLOSER 2014 - Proceedings of the 4th International Conference on Cloud Computing and Services Science, Barcelona, Spain, April 3-5, 2014.*, M. Helfert, F. Desprez, D. Ferguson, F. Leymann, and V. M. Muñoz, Eds. SciTePress, 2014, pp. 598–603. [Online]. Available: http://dx.doi.org/10.5220/0004957305980603

[15] V. Casola, A. D. Benedictis, and M. Rak, "Security monitoring in the cloud: An sla-based approach," in *10th International Conference on Availability, Reliability and Security, ARES 2015, Toulouse, France, August 24-27, 2015*. IEEE, 2015, pp. 749–755. [Online]. Available: http://dx.doi.org/10.1109/ARES.2015.74

[16] B. Wehbi, E. M. de Oca, and M. Bourdellès, "Events-based security monitoring using MMT tool," in *Fifth IEEE International Conference on Software Testing, Verification and Validation, ICST 2012, Montreal, QC, Canada, April 17-21, 2012*, 2012, pp. 860–863.

[17] M. Rak, N. Suri, J. Luna, D. Petcu, V. Casola, and U. Villano, "Security as a service using an sla-based approach via specs," in *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, vol. 2. IEEE, 2013, pp. 1–6.

[18] A. De Benedictis, M. Rak, M. Turtur, and U. Villano, "Rest-based sla management for cloud applications," in *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2015 IEEE 24th International Conference on*. IEEE, 2015, pp. 93–98.

[19] S. O. Afolaranmi, L. E. G. Moctezuma, M. Rak, V. Casola, E. Rios, and J. L. M. Lastra, "Methodology to obtain the security controls in multi-cloud applications," in *CLOSER 2016 - Proceedings of the 6th International Conference on Cloud Computing and Services Science, Rome, Italy, April 23-25, 2016*.