On the Adoption of Security SLAs in the Cloud

Valentina Casola[1](✉), Alessandra De Benedictis[1], and Massimiliano Rak[2]

[1] Universita' di Napoli Federico II, Naples, Italy
{valentina.casola,alessandra.debenedictis}@unina.it
[2] Seconda Universita' di Napoli, Caserta, Italy
massimiliano.rak@unina2.it

Abstract. Can security be provided as-a-Service? Is it possible to cover a security service by a proper Service Level Agreement? This paper tries to reply to these questions by presenting some ongoing research activities from standardization bodies and academia, trying to cope with the open issues in the management of Security Service Level Agreement in its whole life cycle, made of negotiation, enforcement and monitoring phases.

Keywords: Service Level Agreement · Security SLA · Cloud computing · SLA life cycle

1 Introduction

Security still represents one of the main limits in the adoption of cloud computing. It is not rare the case where cloud service providers (CSPs) offer *non-transparent* security mechanisms, embedded in the systems, which are non-negotiable and, above all, vulnerable. The common approach followed by CSPs is a yes/no solution: they provide (or they declare that they provide) the higher security level available with their technological solutions. As a consequence, customers have a very limited set of offerings in terms of security features, often without real grants about the way in which such mechanisms are actually implemented and granted. Nonetheless, it would be desirable to have security being considered exactly as all the other parameters: we want to negotiate security like all other service terms, and we need a way to let users be aware of what kind of security mechanisms are being put in place to protect their data and applications. At the end of the day, we want to offer Security-as-a-Service and, as for all the other services, we want to deliver it under the control of SLAs. The problem is that currently the SLAs are mainly focused on service-related aspects such as performance and availability, and very few terms are related to security (mainly disaster recovery and business continuity). Indeed, many European initiatives have been activated by the European Community to define a common understanding and semantic of SLAs for cloud computing [15], and specific subgroups are working on security-related aspects but, up to date, security is not under negotiation.

The idea of Security-as-a-Service implies that we can dynamically "add security" to services even when they are offered by public, potentially untrusted,

M. Felici and C. Fernández-Gago (Eds.): A4Cloud 2014, LNCS 8937, pp. 45–62, 2015.
DOI: 10.1007/978-3-319-17199-9_2

CSPs. Indeed, to do this, we would need mechanisms that are able to: (i) automatically enforce security mechanisms and controls; (ii) automatically monitor the "security level" promised in the SLA; (iii) face the problem of negotiation, taking in consideration that, typically, users sign the SLAs but, while they understand the concepts of "response time", "availability of a system" or "offered functionalities", they are not security experts, as they usually understand and express very general security needs (e.g., "I want a secure system, always available") but they are not able to translate general terms in actual security mechanisms and controls to enforce.

Indeed, it is up to a security administrator to translate user security requirements, standard guidelines and compliance policies into low-level security mechanisms, policies and configurations, able to cover all security related points that are addressed in those high-level document-based requirements and procedures. So, what are the problems associated to finding and signing an agreement in terms of security requirements? We can face the problem from two different perspectives: from (i) a semantic point of view, there is the need for a common vocabulary to express and evaluate security parameters, while from (ii) an operational point of view, we have to consider automatic mechanisms that from one side are able to enforce proper security mechanisms to meet specific security requirements, and from the other side are able to continuously monitor the security requirements that have been promised.

As for negotiation, there is the need for mechanisms to specify cloud security requirements, and to let users understand the standalone and comparative security features offered by different CSPs. As for the enforcement of security, from a technological point of view, there are no difficulties in enforcing service-based mechanisms once the list of mechanisms, their configurations and how to enforce them are available. As for the monitoring, due to the gap between the actual security mechanisms adopted and user expectations, it is common practice for cloud users to "blindly trust" their CSPs, and to react (e.g., closing subscriptions and changing their provider) only after a security incident has occurred.

How to monitor security and security SLAs when they affect services offered by public, external CSPs? Which are the parameters to monitor? The problem is still open and few dedicated solutions exist. Furthermore, the monitoring problem is even worse in cloud than for traditional outsourcing providers, especially if we take in consideration the different cloud deployment models (IaaS, PaaS and SaaS), where responsibilities are shared among customers and providers in different ways, according to what is offered as-a-service and what is under the control of the customers. Traditional SIEM systems, Intrusion Detection Systems and Vulnerability Assessment tools may not suffice in the cloud.

Despite the state of the art efforts, aiming to build and represent security parameters in cloud SLAs (e.g., the CSA SLA and PLA working groups, or research projects as A4cloud, CUMULUS, Tclouds and Contrail), there are no available user-centric solutions offering systematic mechanisms to manage the whole SLA life-cycle. In this paper, we are going to present a number of international initiatives related to standardization efforts for a common SLA vocabulary and related to research project results, and we will present the original approach provided by the SPECS project [6], whose main goal is just to provide security-as-a-service in the cloud environment, with an approach based on the management of the SLA life cycle. We will focus our attention on one of the main problems that is faced within the project, namely the quantitative evaluation of security. Indeed, we will present two different techniques (namely the Reference Evaluation Methodology and the AHP-based evaluation technique) to evaluate security, starting from the security parameters described in a formalized SLA and how they can be evaluated.

The remainder of the paper is structured as follows: in Sects. 2 and 3 we will provide an overview of the motivation to reasoning about the SLAs and in particular the Security SLAs, and we will discuss a number of initiatives from standardization bodies and from the scientific community towards the Security SLA and the concept of metrics to evaluate security. In Sect. 4, we will present the SLA-based approach provided by the SPECS project and in particular we will present the main features and issues associated to the SLA life cycle management; we will present two techniques to evaluate the security provided by CSP that help in the negotiation phase and we will illustrate some mechanisms and approach toward automatic enforcement and monitoring. Finally, in Sect. 5 some conclusions will be drawn.

2 Reasoning on SLA

As discussed in the Introduction, an SLA is a contract among a provider and its customers stating the quality level of the services offered. In addition to the list of covered services and to the service terms guarantees, an SLA should clearly state how to determine whether the provider is delivering the service as promised or not. Moreover, it should include the responsibilities of both the provider and the consumer of the services, and the remedies to be applied by the provider or the customer in case some terms are not respected. In the practice, this rarely happens in state of art SLAs, which are often legal documents, written in natural language and including the above discussed concepts only in an informal way.

When moving towards automatic reasoning on SLAs, which requires the SLAs to be expressed in a machine readable format, the side effect is that, usually, service providers focus on technical aspects and on what they are actually able to measure, while customers focus on the requirements they have with respect to their usage of the application.

When applying these concepts to security, the complexity grows easily, as there is a *semantic gap* between the customer, which has specific security requirements, but often does not have expertise on security terminology, and the provider, which aims at expressing the security level of its services with respect to very detailed technical terms, with focus on the service behaviour and not on how it will be used by (one of) its customers. The issue becomes even more complex when considering a typical cloud customer, which acquires resources from one or more CSPs in order to build up a cloud application to sell services to other customers.

Such kind of customers have clear responsibilities over the security of the services they offer, but do not have full control over the resources on top of which they run such services. In such scenario, the cloud customer requirements may be specified in a *Security SLA*: as an example, the customer may require that data confidentiality is granted, implying that data must be encrypted both while in motion and at rest, and in such cases the details of the encryption algorithms and access control policies should be specified in the SLA. Moreover, privacy requirements should be taken into account: basic privacy concerns are addressed by requirements such as data encryption, retention, and deletion. An SLA should make it clear how the CSP isolates data and applications in a multi-tenant environment. Even management of data over CSP resources should be clearly stated: how does CSPs prove they comply with retention laws and deletion policies? In cases when regulations must be enforced because of the type of involved data, CSPs should be able to prove compliance. Moreover, for critical data and applications, CSPs should be proactive in notifying customers when the terms of an SLA are violated or at risk, due to infrastructure issues, performance problems and security incidents. Finally, as another example, audit rights should be defined, in order to enable monitoring for any data breaches including loss of data and availability issues. In this case, SLAs should clarify when and how the audits will take place.

Currently, the standard contracts offered by CSPs are one-sided and service provider-friendly, with little opportunity to change terms. Few CSPs offer meaningful service levels or assume some responsibility for legal compliance, security or data protection. Many permit suspension of service or unilateral termination, and disclaim all or most of the provider's potential liability. In this scenario, there is no space for customer requirements. This is contradictory with the cloud computing paradigm, which assumes the *On-demand self-service* as one of the basic characteristics of cloud computing: the customer should be able to select and activate services without any human interaction. The side effect, from an SLA point of view, is that the SLAs should be automated, facing the problems outlined above, trying to adapt the requests to the actual state of the provider resources, in order to grant the respect of agreed terms.

A solution can be provided by filling the semantic gap between CSPs (good practices, guidelines, compliance policies,) and security mechanisms to enforce (technology specific) and monitor to guarantee security levels (Fig. 1). As outlined in Fig. 1, the goal of the Security SLAs should be to fill the gap between the typical mechanisms and solutions adopted in security, the security control models and the cloud architecture.

In order to meet such goal, it is fundamental to have a common and standardized security vocabulary that helps in a clear mapping between requirements, security controls and cloud architecture. As it will be shown in the next section, at the state of art such shared and standard vocabulary does not exist. The second relevant aspect is the capability to quantitatively evaluate the security offering in order to define the *security levels*, which will be addressed in Sect. 4.1.

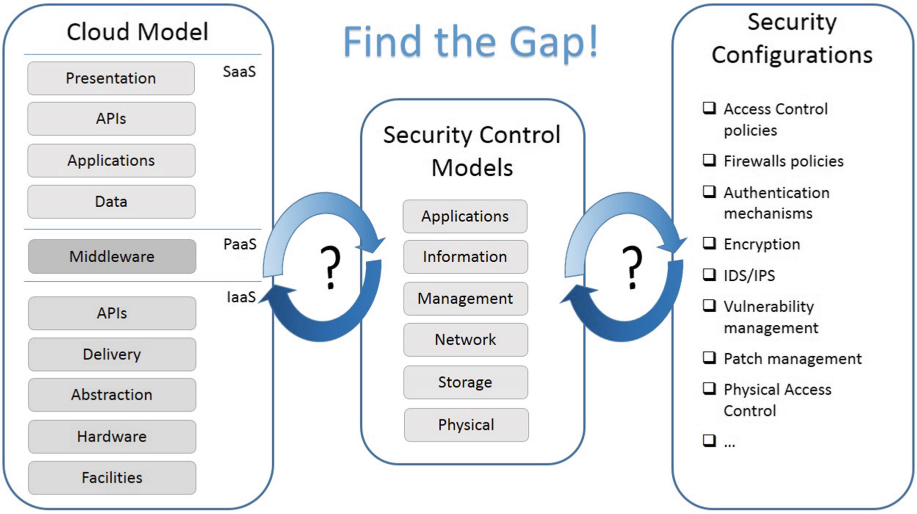


Fig. 1. The semantic gap among security requirements, guidelines and configuration

3 Related Work

As outlined in Sect. 2, the lack of a shared and standard vocabulary for both security and cloud concepts is one of the main limits for the adoption of Security SLAs. Actually, the cloud computing paradigm is relatively young: only in September 2011 the widely accepted cloud definition from NIST [20] was considered definitive, but up to that moment dozen of different definitions were proposed. The state of art of standardization in the cloud context is well outlined by the European Commission in the *Unleashing the Potential of Cloud Computing in Europe* Directive [15]. Currently, two main proposals have been released to define a cloud reference architecture, namely the *NIST Cloud Reference Architecture* [19] and the *ISO Reference Architecture and Vocabulary* [1,2].

For what regards the adoption of SLAs, the state of art is in a similar condition: the only (de facto) standard for a machine readable format is WS-Agreement [5], born in the GRID context, while more high-level standards aiming at defining in detail what an SLA should contain are the *ISO 19086* [3], still not definitive, and the initiative from the European Commission, i.e., the Cloud Selected Industry Group on Service Level Agreement (C-SIG SLA), which released a guideline for the production of standards on SLAs [23].

As for the security field, the main reference for security controls definition and best practices is the set of standards proposed by NIST and ISO. In particular, the NIST 800-53 and 800-53A documents offer guides for assessing security controls and ISO recommendations 20000-1, 20000-7, 27036-4, 27001, 27002, 27017 and 27018 comprise aspects such as service management of cloud services, guidelines for suppliers regarding Information Security, requirements and metrics for security management also applied to cloud computing. ISO has recently started an initiative in order to identify specific security controls for the cloud context (ISO 27017).

For the cloud specific environment, it is relevant to outline the role of Cloud Security Alliance, which provided set of questionnaires to evaluate the security level of cloud providers through a three layer approach to Security Certification named STAR, whose starting is a publicly available self-assessment questionnaire (CAIQ) [14]. In such context, CSA proposed a Cloud Control Matrix (CCM) [13] which lists the possible security controls to be adopted in the cloud context.

From the research activities point of view, the cloud security and Security SLA problem is addressed directly and indirectly in many active research projects. As an example, Accountability for Cloud (A4Cloud[1]) aims to improve the acceptability of cloud-based infrastructures where critical data is perceived to be at risk. CUMULUS[2] develops an integrated framework of models, processes and tools to support the certification of security properties of multi-layer cloud services using multiple types of evidence for security, including service testing, monitoring data and trusted computing proofs. CIRRUS[3] is one more project focusing mostly on certification and standardization in cloud. In the cloud context, CONTRAIL [18] is an IP project that addresses, among a lot of other issues, the SLA management in Cloud Federations, topic that was addressed even in mOSAIC[4]. Finally, TClouds[5] is the first EC-ICT project that deeply analyzes the security issues. It has until now delivered a set of solutions belonging to the IaaS level.

4 The SLA-Based Approach to Cloud Security

Starting from the previous consideration, we are going to propose an innovative approach to provide security services in the cloud that are offered under the control of SLAs with security parameters. At this aim, we need to be able to manage an SLA in its whole life cycle and to automatically enforce security by implementing security mechanisms that cover the desired security parameters.

Figure 2 illustrates a simplified view of the SLA life cycle management with a flow chart diagram, which is mainly based on five different activities:

- *negotiation:* to cope with users and providers needs and requirements and find an agreement;
- *enforcement:* to implement proper security mechanisms and controls;
- *monitoring:* to be able to continuously monitor the security parameters to guarantee;
- *remediation:* to provide proper actions in case of some alert conditions;
- *re-negotiation:* to change the SLA in case some security provision has been violated.

[1] The A4Cloud project web site, http://www.a4cloud.eu/.

[2] The CUMULUS project web site, http://www.cumulus-project.eu/.

[3] The CIRRUS project web site, www.cirrus-project.eu.

[4] The mOSAIC project web site, http://www.mosaic-cloud.eu/.

[5] The TClouds project web site, http://www.tclouds-project.eu/.

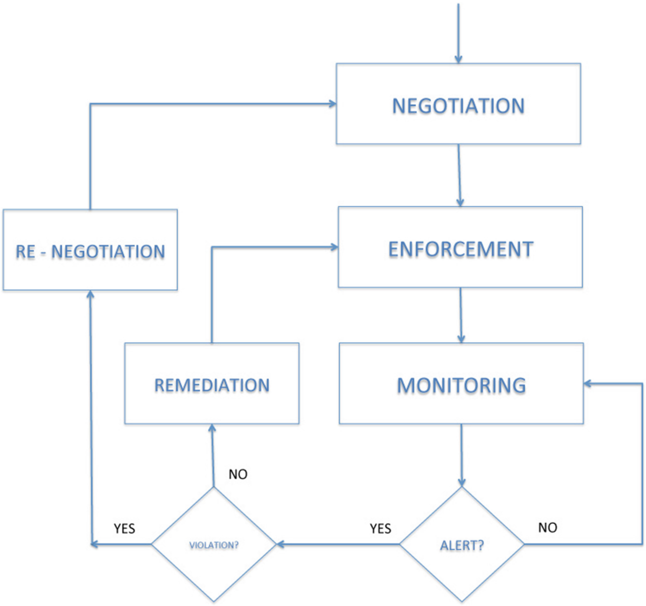


Fig. 2. The SLA life cycle management

In particular, in the negotiation phase there is the need for tools that support users to express their security requirements, and for methodologies to evaluate the security provided by different providers. In the enforcement phase, there is the need to properly plan the activation of security mechanisms to be able to build secure service invocations. In the monitoring phase, there is the need to guarantee security by continuously monitoring security parameters. Remediation and re-negotiation apply only when some alert or violation conditions occur, and should be managed properly. Indeed, the SPECS (Secure Provisioning of cloud Services based on SLA management) project [6] goals are mainly related to cope with these problems in order to provide and control security through proper services. In fact, SPECS aims at supporting both cloud customers and CSPs to respectively access and provide a secured target service. SPECS Security Services provide security guarantees to cloud customers, by using specific services to negotiate, enforce and continuously monitor the security parameters included in the SLA (SLA-based approach).

In next subsections, we are going to provide some in-sight on the research ongoing activities on negotiation, and in particular on the methodologies to evaluate the security provided by a CSP, on the enforcement and on the monitoring of security services, giving an overview on what is actually available in the literature and what are the future works to be done to effectively implement the SLA-based approach.

4.1 Security SLA Negotiation: Evaluating Security

As already said, of the main activities associated to the negotiation process is the assessment and evaluation of security. In this section, we will focus our analysis on two security evaluation methodologies to express and evaluate security terms included in an SLA [10], namely the Reference Evaluation Model (REM) [11] and an AHP-based evaluation methodology [8]. Thanks to these, we are able to (i) express security through a semi-formal and not ambiguous model (Security SLA) where the chosen formalization is easy to adopt for both customers and security experts; (ii) evaluate the security level that a security service is able to guarantee by aggregating the security associated to all SLA security terms (multi-decision approach); (iii) evaluate/compare/rank different providers offering different systems according to the measured quality/security level.

The Reference Evaluation Methodology (REM). The first methodology that we present is the Reference Evaluation Model (REM) [9,12], whose goal is to provide an automatic means to state the security level provided by a service. The methodology defines how to express in a rigorous way the security SLA, how to evaluate a formalized SLA, and how to state the provided security level. Any SLA is represented through a tree, which contains all the SLA security terms (intermediate nodes and leaves). In Fig. 3 the three methodology phases are shown: **Policy Structuring**, **Policy Formalization** and **Policy Evaluation**:

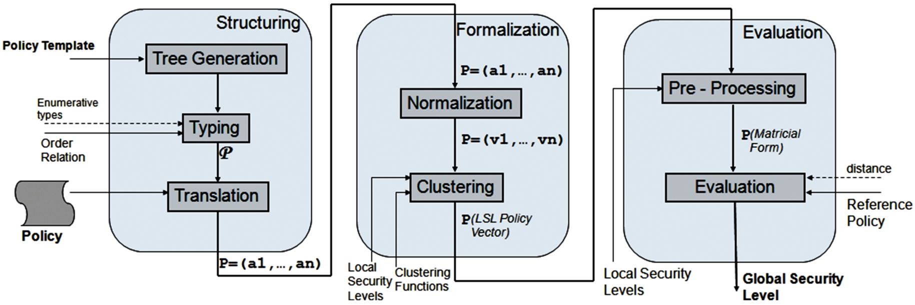


Fig. 3. Phases of the evaluation methodology

1. The goal of the **Structuring** phase is to associate an enumerative and ordered data type K_i to the n leave-security terms of the SLA. An SLA space "P" is defined as P $=K_1 \times K_2 \times \ldots \times K_n$, i.e., the vector product of the n security terms K_i. The space is defined according to an SLA template that strongly depends on the application context.
2. The main goal of the **Formalization** phase is to turn the SLA space "P" into a homogeneous space "PS". This transformation is accomplished by a normalization and clusterization process which allows to associate a Local Security Level (LSL) to each provision; after that, the security terms may be compared by comparing their LSLs.

3. The main goal of the **Evaluation** phase is to pre-process the "PS" vector of LSLs in order to represent it by a $n \times 4$ matrix whose rows are the single security terms K_i and the number of columns is the chosen number of LSLs for each provision. For example, if the number of LSL is four and the LSL associated to a provision is l_2, the row in the matrix associated to the provision in the matrix will be: (1,1,0,0). Finally, a distance criteria for the definition of a metric space is applied. REM adopts the Euclidean distance among matrices:
   $d(A, B) = \sqrt{(\sigma(A - B, A - B))}$
   where $\sigma(A - B, A - B) = Trace((A - B)(A - B)^T)$

To define the Global Security Level L_{Px} associated to the SLA P_x, we have introduced some reference levels and adopted the following metric function:

$$L_{Px} = \begin{cases} L_0 \ iff d_{x0} \leq d_{10} \\ L_1 \ iff d_{10} < d_{x0} < d_{20} \\ L_2 \ iff d_{20} < d_{x0} < d_{30} \\ L_3 \ iff d_{30} < d_{x0} < d_{40} \\ L_4 \ iff d_{40} \leq d_{x0} \end{cases}$$

where $d_{i,0}$ are the distances among the references and the origin of the metric space (denoted as $\emptyset$). This function gives a numerical result to the security; the idea is to evaluate the security associated to a service through the evaluation of its security SLA.

The GSL is a measure of the security provided by an service according to its security SLA; it is obtained by formalizing the process that is manually performed by security experts while trying to extend trust to other domains. The details of the methodology are out of the scope of this paper, and they can be found in [9].

The AHP Based Methodology. To satisfy the flexibility, adaptability, and interoperability requirements that are necessary in SLA stipulation and monitoring processes, we propose to formalize SLA policies according to hierarchical Quality Models whose structure must be defined according to the rules of the following Quality meta-model [7].

The SLA Quality meta-model comprehends some fundamental concepts:

- Quality Characteristic: any quality requirements, such as Performance, Security, Cost, Maintainability
- Characteristics may be arranged in a hierarchy (Measurable Characteristics are the leaves)
- Measurable Characteristic: a Quality Characteristic that can directly be measured

In Fig. 4 we reported the quality meta-model and we formally express service SLAs as an instance of the meta-model.

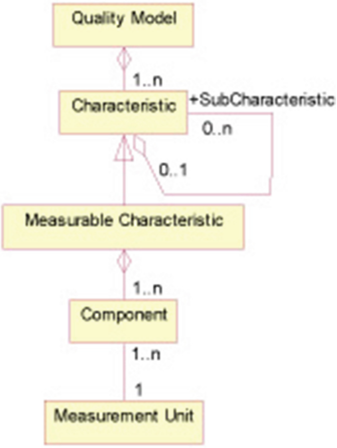


Fig. 4. The SLA quality meta-model

The evaluation process is made of two different steps:

1. A security expert *designs the decision model*;
2. The decision maker evaluates the quality by *applying the decision model.*

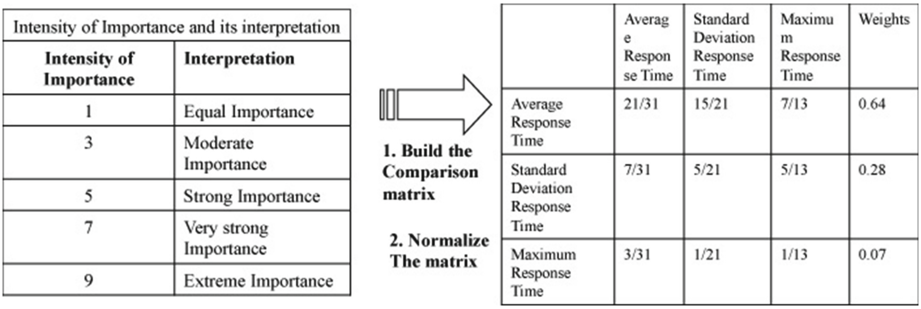
The **decision model design activity** includes three main steps, i.e., Weight Assignment, Clustering, and Rating. They are preliminarily performed just once, independently of the number of evaluations that will be performed. In the Weight Assignment step, the relative importance of the characteristics is rated; in the Clustering step, for each measurable characteristic, the sets of values that will be considered equivalent for the aims of the evaluation are defined; finally, in the Rating Step, each set is associated to a rating value. In the following, an example of these three steps is reported:

Step 1: Weight Assignment. For each Characteristic not directly measurable, the decision process designer will estimate the relative *Intensity of Importance* of any pair of its n Sub-Characteristics, by defining a matrix of n*n. Then, the designer has to build the *Comparison matrix* and then *Normalize* the matrix to define security parameters weights, as illustrated in Fig. 5.

Steps 2 and 3: Clustering e Rating. To cluster the possible values of an SLA offering, the designer has to define a *Utility Function* R to order the possible values on the basis of relative (and not absolute) preferences (as the Local Security Levels of the REM methodology): given two values x and y of the set, if x is preferred to y then $R(x) > R(y)$. Let us consider, as an example, the *Average Response Time* Characteristic, we can define the utility function as follows:

$$R = Offered_{value}/Requested_{value} \tag{1}$$

Then, all possible solutions to this function are clustered in three levels according to this meaning: {very fast response, sufficiently fast response, quite slow response}:



Intensity of Importance and its interpretation	
Intensity of Importance	**Interpretation**
1	Equal Importance
3	Moderate Importance
5	Strong Importance
7	Very strong Importance
9	Extreme Importance

1. Build the Comparison matrix

2. Normalize The matrix

	Average Response Time	Standard Deviation Response Time	Maximum Response Time	Weights
Average Response Time	21/31	15/21	7/13	0.64
Standard Deviation Response Time	7/31	5/21	5/13	0.28
Maximum Response Time	3/31	1/21	1/13	0.07

Fig. 5. Weight assignment

$$UF = \begin{cases} R < 0.5 \\ 0.5 <= R < 1 \\ 1 <= R < 2 \end{cases} \tag{2}$$

After clustering each possible value, the designer rates all clusters according to their *Goodness* (the Goodness of a cluster is defined in the same way as the weight) and defines the following *Satisfaction Function* that represents the relative rate/evaluation of a cluster:

$$Ssc(R) = \begin{cases} 0.63 \; iffR < 0.5 \\ 0.26 \; iff0.5 <= R < 1 \\ 0.11 \; iff1 <= R < 2 \end{cases} \tag{3}$$

In the *decision making activity*, a customer can easily compare the security of an offered service (expressed in the Quality Offer Model) against his needs (expressed in the Quality Request Model); the security of different services is compared by evaluating:

1. a *Satisfaction Function* for each Measurable Characteristic;
2. a *Satisfaction Function* for each non-Measurable Characteristic;
3. the *Overall Satisfaction Function* that aggregates all evaluations.

In particular, for each Characteristic c, the *Satisfaction Function* can be evaluated as the weighted sum of the Satisfaction of its Sub-Characteristics:

$$S_C(req.off) = \sum\nolimits_{sc \in C(c)} w_{SC} S_{SC}(req, off) \tag{4}$$

where $C(c)$ is the set of Sub-Characteristics affecting the Quality Characteristic; w_{SC} is the weight of the Sub-Characteristic sc and $S_{SC}(req, off)$ is the value of the Satisfaction Function of the Sub-Characteristic sc.

Finally, the *Overall Satisfaction* of a service offering is given by:

$$S(req.off) = \sum\nolimits_{c \in C} w_c S_c(req, off) \tag{5}$$

where the set C includes all the higher level Characteristics of the customer Quality Model, while wc is the weight of that Characteristics and $S_c(req, off)$

is the Satisfaction value of the Characteristic *c*. We invite the interested reader to refer to [7] for a more detailed discussion of this technique.

The two proposed techniques have many points in common: indeed, the security terms and associated metrics are closely connected with services, system parameters and configuration mechanisms, and they are always pre-evaluated by security experts in order to let the evaluation process be automatic. Among the positive aspects, they always take into account both the user requirements and evaluators perspectives (weight definition, cluster assignment, security level definition). As for the drawbacks, the security is evaluated starting from a static evaluation of enforced security mechanisms (but the security provided changes with time due to new attacks and vulnerability), and there is the need for automatic monitoring systems to assure the respect of security parameters. Once the different providers' offerings are evaluated, the customer can choose the one that best fits his needs and, finally, he can sign the SLA. Moreover, there is the need for automatic mechanisms to enforce security mechanisms and, possibly, redress security mechanisms to react to some alert and there is the need for monitoring systems that are targeted to security parameters. In the next two sections we will present an overview of current open issues and available solutions on these last points, too.

4.2 Security SLA Enforcement

After the negotiation phase, a signed SLA must be implemented. SLA implementation involves making sure that the negotiated service levels are correctly set up and monitored, in order to report possible failures and trigger proper reactions. SLA monitoring will be discussed in the next subsection, while in the following we provide an overview of the issues related to the configuration and activation of a negotiated service fulfilling specific security requirements, and to the management of possible violations.

The basic requirement to provide cloud end-users with the services they requested in the negotiation phase, is the availability of security mechanisms, protocols and tools offered according to an as-a-service approach. Indeed, in non-trivial cases where the target services are not yet offered with the desired guarantees by any provider, such *security services* may be integrated into the target services' supply chains (i.e., the chains of service invocations involved in the provisioning of the target services), to add the missing features.

Several security-as-a-service products are being currently offered by some vendors, which mainly deal with identity and access management, intrusion detection, encryption etc. In the context of research projects, we can mention the Consec framework defined within the Contrail project, aimed at providing a stack of software components for the federation of independent clouds. ConSec is a framework that enables any infrastructure provider to make use of federated (i.e., external) identity management with authorization and auditing features. This solution may be useful to enrich the catalog of security services available for the enforcement of SLOs with specific pluggable services for identity federation, authorization and auditing.

While the identification of proper supply chains able to guarantee the requested secure services is a task to be accomplished during the negotiation phase (to determine whether a solution exists to satisfy the end-user requests), the actual set up of the supply chain acknowledged by the end-user is performed during the SLA implementation. It involves retrieving, configuring and activating all services/resources needed to have the desired service up and running, and requires a planning activity aimed at defining a complete workflow. Once set up, the services should be monitored in order to detect possible violations of the signed SLA, which would imply the application of penalties or other actions taken against the provider. In this scenario, from the perspective of both the customer and the provider, it would be desirable to be warned about a possible incoming violation, in order to take proper countermeasures and avoid it. At this aim, enforcement should also envision a diagnosis activity, for the analysis of monitoring data, and include the capability of identifying, if possible, the best reaction strategy to implement.

An example of security management framework aimed at mediating between cloud services and security mechanisms is described in [4]. The proposed framework is composed of three layers, namely the management layer, the enforcement layer, and the feedback layer, which are respectively responsible of: (i) defining the security specifications of the cloud service providers and customers, (ii) planning security and selecting security controls based on identified risks, and (iii) collecting and analyzing measurements related to security metrics to ensure that the system is operating within the defined boundaries, by triggering configuration updates in case of deviations from the defined boundaries.

The described approach is followed by other projects, which also adopt SLAs for security specification and assurance. As an example, we mention SLA@SOI[6], that proposes a solution for orchestrating services on the basis of SLAs. In SLA@SOI, the enforcement of the quality characteristics is tightly connected to an SLA manager, a complex software component in charge of the whole management of SLAs.

The SPECS project[7] is also working on improving cloud services' security by adopting an SLA-based approach. The SPECS framework is aimed at managing the whole SLA life-cycle. In particular, the enforcement of SLAs is addressed by a complex module which includes, on the one hand, all components needed to plan and realize the SLA implementation, to reason on monitoring data for diagnosis purposes and to react in case of alerts or violations, and, on the other hand, a catalogue of security services available to improve the security provided by third-parties.

4.3 Security SLA Monitoring

When talking about security monitoring, many questions and open issues should be addressed. As outlined in this section, the problem should be faced by many

[6] The SLA@SOI project web site, http://sla-at-soi.eu/.

[7] The SPECS project web site, http://specs-project.eu/.

point of views. In particular, any monitoring solution should cope with the following questions: *What to monitor?* Physical resource? Physical infrastructures? Or even Virtual Machines and related Software assets? *Where are the monitoring agent?* Many options are configurable in the cloud (monitoring on-premises, monitoring on hosting IaaS, monitoring via SaaS or via other third parties), so which is the configuration that best fits the signed SLA? and *What data* should be monitored? How to manage the huge amount of data? Last, but not least, which security metrics to monitor?

Cloud monitoring typically involves dynamically tracking the Quality of Service (QoS) parameters related to virtualized resources (e.g., VM, storage, network, appliances, etc.), the physical resources they share, the applications running on them and the hosted data. The continuous monitoring of the cloud and of its SLAs, mostly expressed in terms of performance-related guarantees, is of paramount importance for both cloud providers and customers. As for providers in particular, they both aim at preventing SLA violations to avoid penalties, and at ensuring an efficient resource utilization to reduce costly maintenance.

While several tools exist for performance and QoS monitoring in cloud environments, both open source and commercial, security-related monitoring tools are less developed, and current monitoring infrastructures lack appropriate solutions for adequate SLA monitoring. As for security monitoring, few tools exist (many of them are represented by research results), which are typically represented by intrusion detection systems. Therefore, covering all aspects of cloud security SLA monitoring necessarily requires a combination of several monitoring tools.

Most of the current cloud monitoring tools is focused on specific aspects of cloud operation, providing only a partial solution for the cloud monitoring problem. For example, the open source tool Nagios[8] offers complete monitoring and alerting for servers, switches, applications, and services, while Ganglia[9] is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids, which collects dozens of system metrics related to CPU, memory, disk, network and process data. Other examples of popular monitoring tools are the commercial Amazon CloudWatch[10], AzureWatch[11] and OPNET[12].

All mentioned tools are general purpose and have not been designed to directly cope with SLAs. Examples of SLA-oriented monitoring tools are represented by CloudComPaaS[13], LoM2HiS [16] and CASViD [17] or the solution proposed in mOSAIC [22]. CloudComPaaS is an SLA-aware PaaS for managing a

[8] Nagios - The Industry Standard In IT Infrastructure Monitoring, http://www.nagios.org/.

[9] Ganglia Monitoirng System, http://ganglia.sourceforge.net/.

[10] Amazon CloudWatch, http://aws.amazon.com/cloudwatch.

[11] AzureWatch, www.paraleap.com/azurewatch.

[12] OPNET, www.opnet.com.

[13] GRyCAP CloudComPaaS, http://www.grycap.upv.es/compaas/about.html.

complete resource lifecycle, and features an extension of the WS-Agreement SLA specification for cloud computing. The monitor module performs the dynamic assessment of the QoS rules from active SLAs. The three basic operations of the monitor are updating the SLA terms state, checking the guarantees state and performing self-management operations. SLAs registered in the monitor are set to be updated every certain period of time, commonly defined as monitoring cycle. The monitor evaluates the formulas of the guarantee terms and sets the value of the guarantees to either Fulfilled or Violated. CASViD (cloud application SLA violation detection) [17] aims at monitoring and detecting SLA violations at the application layer, and includes tools for resource allocation, scheduling, and deployment. It is an SNMP-based monitoring approach for SLA violation. Service requests are placed through a defined interface to the front-end node, acting as the management node. The VM configurator sets up the cloud environment by deploying pre-configured VM images. The request is received by the service interface and delivered to the SLA management framework for validation, then it is passed to the application deployer for resource allocation and deployment. CASViD monitors the application and sends information to the SLA management framework for detection of SLA violations.

Enabling SLA Monitoring. The first challenge to face to enable security SLA monitoring is to provide a mapping between the application-level security SLOs specified in an SLA and the related measurable low-level metrics. For instance, let us consider the *availability* high-level SLO referred to a cloud application. The application is actually running on physical or virtual resources, which are characterized by low-level metrics such as CPU, memory, uptime, downtime, etc., which are those actually measurable. Thus, there is a gap between the low-level resource metrics and the high-level SLA parameters.

According to ENISA [21], the security parameters for a security monitoring framework can be classified as in Fig. 6. For each parameter, the monitoring and testing methodology, as well as the related thresholds to trigger events (e.g., incident reports or response and remediation) have to be defined. In terms of security requirements, the monitoring tests are quite complex. One of the reasons is the restricted access to the monitoring data, represented in Fig. 7.

Once the security parameters to monitor have been defined, it is necessary to determine appropriate monitoring intervals at the application level, keeping the balance between the early detection of possible SLA violations and the intrusiveness of the monitoring tools on the whole system. With the monitoring of the cloud infrastructure resources, the provider gains information about the usage of the resources and the current resource availability status. The rate of acquiring this information is an important factor influencing the overall performance of the system and the profit of the provider. On the one hand, monitoring at a high rate delivers fast updates about the resource status to the provider, but it can results in a high overhead, which eventually degrades the performance of the system. On the other hand, monitoring at a low rate causes the miss of information such as missing to detect SLA violation, which results in paying

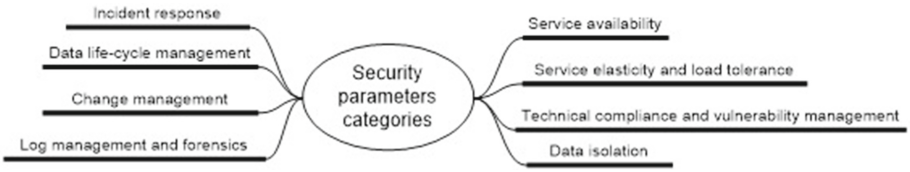

Security parameters categories
Incident response
Data life-cycle management
Change management
Log management and forensics
Service availability
Service elasticity and load tolerance
Technical compliance and vulnerability management
Data isolation

Fig. 6. ENISA security parameters

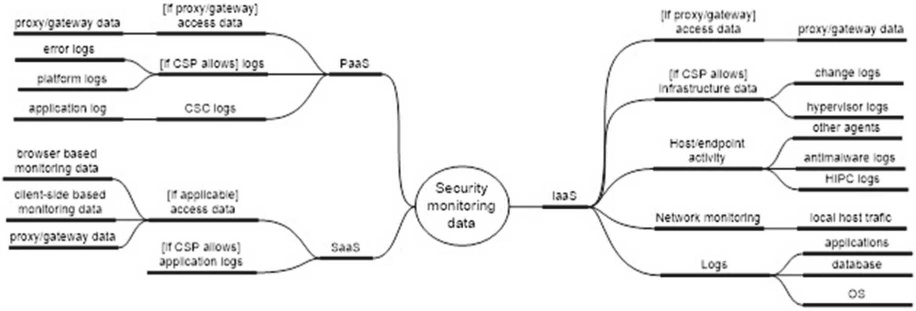


Fig. 7. Security monitoring data

of SLA penalties by the provider. Therefore, to address this issue, techniques to determine the optimal measurement intervals to efficiently monitor to detect SLA violations are required.

A key issue related to the selection of the parameters to monitor is the monitoring granularity. Three main options are possible: client-oriented monitoring, virtual system monitoring and physical system monitoring. Finally, another related issue is the approach adopted to gather monitoring data. Again, three options are possible: use proper APIs offered by the public cloud providers themselves to collect logs, install custom monitoring agents on the monitored infrastructure, or use third-party tools able to gather information on the services under monitoring from the outside.

The adoption of the best configuration of monitoring system to activate, should be automatically related to the security parameters included in the SLA and affect both infrastructures that host many user virtual resources (multitenancy) and user-specific resource to protect. Indeed, the SPECS project is trying to cope with this problem by defining during the SLA enforcement phase the number and typology of monitoring services to activate according to the specific security mechanisms and controls to activate.

5 Conclusions

A Service Level Agreement is a contract among the customer and the provider that states the quality level of the services offered. The scientific and industrial communities are recently investigating the possibility to offer Security-as-a-Service and, above all, to provide such kind of services under specific guarantees formalized in proper SLAs, as done for other services. To reply to this question, many open issues should be addressed, that can be summarized as: (i) it is difficult to express security requirements, (ii) it is difficult to evaluate security and (iii) it is difficult to monitor and guarantee security. These three points represent the main limitation to adopt security SLAs and, even worse, in the cloud environment. In this paper, we tried to address the main research initiatives that face these problems and even presented the SLA-based approach proposed by the SPECS project. In particular, we illustrated different techniques to quantitatively evaluate security, and automatically enforce and monitor the security of cloud services.

Acknowledgment. This research is partially supported by the grant FP7-ICT-2013-11-610795 (SPECS).

References

1. ISO/IEC 17788:2014. Information Technology-Cloud Computing-Overview and Vocabulary. Technical report, International Organization for Standardization (2014)
2. ISO/IEC 17789:2014. Information Technology-Cloud computing-Reference architecture. Technical report, International Organization for Standardization (2014)
3. ISO/IEC NP 19086–1. Information Technology-Cloud computing-Service level agreement (SLA) framework and technology-Part 1: Overview and concepts. Technical report, International Organization for Standardization (2014)
4. Almorsy, M., Ibrahim, A., Grundy, J.: Adaptive security management in saas applications. In: Nepal, S., Pathan, M. (eds.) Security, Privacy and Trust in Cloud Systems, pp. 73–102. Springer, Heidelberg (2014)
5. Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M.: Web Services Agreement Specification (WS-Agreement). Technical report, Global Grid Forum, Grid Resource Allocation Agreement Protocol (GRAAP) WG, September 2005
6. Casola, V., De Benedictis, A., Rak, M., Villano, U.: Preliminary design of a platform-as-a-service to provide security in cloud. In: CLOSER 2014 – Proceedings of the 4th International Conference on Cloud Computing and Services Science, pp. 752–757, Barcelona, Spain, April 3–5 (2014)
7. Casola, V., Fasolino, A.R., Mazzocca, N., Tramontana, P.: An ahp-based framework for quality and security evaluation. In: Proceedings of the 2009 International Conference on Computational Science and Engineering, CSE 2009, vol. 03, pp. 405–411. IEEE Computer Society, Washington, DC (2009)
8. Casola, V., Fasolino, A.R., Mazzocca, N., Tramontana, P.: A policy-based evaluation framework for quality and security in service oriented architectures. In: Proceedings – 2007 IEEE International Conference on Web Services, ICWS 2007, pp. 1181–1182 (2007)
9. Casola, V., Mazzeo, A., Mazzocca, N., Rak, M.: An innovative policy-based cross certification methodology for public key infrastructures. In: Chadwick, D., Zhao, G. (eds.) EuroPKI 2005. LNCS, vol. 3545, pp. 100–117. Springer, Heidelberg (2005)

10. Casola, V., Mazzeo, A., Mazzocca, N., Rak, M.: A sla evaluation methodology in service oriented architectures. In: Gollmann, D., Massacci, F., Yautsiukhin, A. (eds.) Advances in Information Security, pp. 119–130. Springer, USA (2006)
11. Casola, V., Mazzeo, A., Mazzocca, N., Vittorini, V.: A policy-based methodology for security evaluation: a security metric for public key infrastructures. J. Comput. Secur. **15**(2), 197–229 (2007)
12. Casola, V., Mazzocca, N., Luna, J., Manso, O., Medina, M., Rak, M.: Static evaluation of certificate policies for grid pkis interoperability. In: Proceedings - Second International Conference on Availability, Reliability and Security, ARES 2007, pp. 391–399 (2007)
13. Cloud Security Alliance. Cloud Control Matrix v3.0. https://cloudsecurityalliance.org/download/cloud-controls-matrix-v3/
14. Cloud Security Alliance. Consensus Assessment Initiative Questionnaire V1.1. https://cloudsecurityalliance.org/research/cai/
15. European Commission. SWD(2012) 271 final. Unleashing the Potential of Cloud Computing in Europe. Technical report, September 2012
16. Emeakaroha, V.C., Brandic, I., Maurer, M., Dustdar, S.: Low level metrics to high level slas - lom2his framework: bridging the gap between monitored metrics and sla parameters in cloud environments. In: 2010 International Conference on High Performance Computing and Simulation (HPCS), pp. 48–54, June 2010
17. Emeakaroha, V.C., Ferreto, T.C., Netto, M.A.S., Brandic, I., De Rose., C.A.F.: Casvid: Application level monitoring for sla violation detection in clouds. In: 2012 IEEE 36th Annual Computer Software and Applications Conference (COMPSAC), pp. 499–508, July 2012
18. Harsh, P., Jegou, Y., Cascella, R.G., Morin, C.: Contrail virtual execution platform challenges in being part of a cloud federation. In: Abramowicz, W., Llorente, I.M., Surridge, M., Zisman, A., Vayssière, J. (eds.) ServiceWave 2011. LNCS, vol. 6994, pp. 50–61. Springer, Heidelberg (2011)
19. Liu, F., Tong, J., Mao, J., Bohn, R.B., Messina, J.V., Badger, M.L., Leaf, D.M.: NIST SP - 500–292. Cloud Computing Reference Architecture. Technical report, National Institute of Standards & Technology, September 2011
20. Mell, P.M., Grance, T.: SP 800–145. The NIST Definition of Cloud Computing. Technical report, National Institute of Standards & Technology, Gaithersburg, MD, United States (2011)
21. European Network and Information Security Agency (ENISA). Procure secure. a guide to monitoring of security service levels in cloud contracts, April 2012
22. Rak, M., Venticinque, S., Mahr, T., Echevarria, G., Esnal, G.: Cloud application monitoring:the mosaic approach. In: 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), pp. 758–763, November 2011
23. European Commission C-SIG (Cloud Select Industry Group) subgroup. IP/14/743. New guidelines to help EU businesses use the Cloud. Technical report, June 2014

http://www.springer.com/978-3-319-17198-2

Accountability and Security in the Cloud
First Summer School, Cloud Accountability Project, A4Cloud,
Malaga, Spain, June 2-6, 2014, Revised Selected Papers
and Lectures
Felici, M.; Fernández-Gago, C. (Eds.)
2015, X, 307 p. 44 illus., Softcover
ISBN: 978-3-319-17198-2