2013 IEEE International Conference on Cloud Computing Technology and Science

Security as a Service Using an SLA-based Approach via SPECS

Massimiliano Rak*, Neeraj Suri†, Jesus Luna†, Dana Petcu ‡,Valentina Casola§ and Umberto Villano¶
*Second University of Naples, Italy
Email: massimiliano.rak@unina2.it
†Technical University Darmstadt, Germany
Email: suri@cs.tu-darmstadt.de, jluna@deeds.informatik.tu-darmstadt.de
‡Institute e-Austria Timisoara, Romania
Email: petcu@info.uvt.ro
§University of Naples Federico II , Italy
Email: casolav@unina.it
¶University of Sannio, Italy
Email:villano@unisannio.it

***Abstract*—The cloud offers attractive options to migrate corporate applications, without any implication for the corporate security manager to manage or to secure physical resources. While this ease of migration is appealing, several security issues arise: can the validity of corporate legal compliance regulations still be ensured for remote data storage? How is it possible to assess the Cloud Service Provider (CSP) ability to meet corporate security requirements? Can one monitor and enforce the agreed cloud security levels? Unfortunately, no comprehensive solutions exist for these issues. In this context, we introduce a new approach, named SPECS. It aims to offer mechanisms to specify cloud security requirements and to assess the security features offered by CSPs, and to integrate the desired security services (e.g., credential and access management) into cloud services with a Security-as-a-Service approach. Furthermore, SPECS intends to provide systematic approaches to negotiate, to monitor and to enforce the security parameters specified in Service Level Agreements (SLA), to develop and to deploy security services that are cloud SLA-aware and are implemented as an open-source Platform-as-a-Service (PaaS). This paper introduces the main concepts of SPECS.**

I. Introduction

While the economic and technological advantages of cloud computing are appealing, the migration of key sector applications onto the cloud has been partly limited by the lack of comprehensible and enforceable security assurance provided by Cloud Service Providers (CSP).

The typical cloud user is not a security expert; nevertheless, he has some security requirements to fulfill (e.g., due to regulatory compliance) that are usually expressed in an informal manner. For such a user, managing his/her security requirements (e.g., negotiating, monitoring and enforcing) on the top of the security offered "natively" by a CSP is a task that is both expensive and unrealistic to accomplish. Due to this gap between reality and expectations, it is common practice for cloud users to "blindly trust" their CSPs, and to react (e.g., changing their provider) only after a security incident has occurred. This problem worsens if we take into account the ever-increasing number of CSPs available in the cloud ecosystem. Moreover, a number of natural concerns arise. Despite the assumption that a given CSP "seems" secure, is it actually secure enough for a given application? How is it possible to compare multiple CSPs with regards to security?

Even if the problem is particularly challenging, security assurance in cloud computing recently has taken some initial but promising steps. The cloud security community, represented by workgroups at the European Network and Information Security Agency (ENISA) and the Cloud Security Alliance (CSA), has identified that specifying security parameters in Service Level Agreements (SLAs) is useful to establish a common semantic to manage cloud security from two perspectives. These are: (i) the security level being offered by a CSP, and (ii) the security level requested by a cloud user.

Despite current efforts, aiming to build and to represent security parameters in cloud SLAs (e.g., the CSA SLA and PLA working groups [14], research projects as A4Cloud, CUMULUS, TClouds and Contrail), there are no available user-centric solutions (i.e., empowering cloud customers) offering systematic mechanisms to manage their whole life-cycle.

CSPs currently do not offer any rigorous specification of SLAs that describes formally their security features. Furthermore, there is a dearth of tools that could help cloud end-users to negotiate, to monitor and to enforce the security features they need. As pointed out by the ENISA report on cloud SLAs [15], "the survey data shows that while SLAs are often used, and availability is often addressed in these SLAs, other security parameters are less well covered . . . ".

To respond to the cloud community needs, the SPECS project has been initiated, in the frame of the FP7-ICT programme. It intends to improve the state-of-the-art in cloud computing security by creating, promoting and exploiting a user-centric framework and a platform dedicated to offer Security-as-a-Service by an SLA-based approach, in particular with respect to negotiation, continuous monitoring and enforcement. This paper aims at describing the SPECS vision of the cloud security problem, and to describe the approach explored to offer a clear and flexible solution to (some of) the common problems related to cloud security. In particular, the management of SLA in the whole life-cycle will enable to face security requirements engineering for cloud computing with a different prospective, i.e., being able to negotiate and evaluate enforced security solutions.

978-0-7695-5095-4/13 $31.00 © 2013 IEEE
DOI 10.1109/CloudCom.2013.165

The remainder of this paper is organized as follows. The next section aims at outlining the cloud security problem, while Section III describes the approach proposed within the SPECS project to tackle this issue. Section IV describes the usage scenarios of the proposed framework, and Section V summarizes the state-of-the-art and outlines the advances that SPECS proposes. Finally, Section VI draws the conclusions and sketches the SPECS ongoing activities.

II. Security-as-a-Service

In the vision of the common user, security issues refer to a very large set of different problems, depending on the context, on the people and the resources involved. Despite this, it is common to refer to security problems as a whole, aiming at facing all of them together. The cloud computing approach, which collects under the same umbrella an incredible number of different technologies and approaches, makes worse the perception of lack of security, being security *the requirement* that can be hardly met and monitored.

Common security solutions are usually based on the location and hardening of vulnerabilities, through the adoption of proper security mechanisms to meet specific requirements. In cloud computing, the adoption of virtualization, of additional abstraction layers, and the flexibility of the infrastructure make the vulnerability analysis very difficult and, consequently, the application of well known security techniques (network security, protection against denial of service, access control, ...) cannot be sufficient.

Such consideration pushes a re-thinking of security to take into account the five main cloud characteristics, as introduced by NIST: on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service. A recent approach to match all these requisites is to offer security *as-a-service*. The Cloud Security Alliance recently proposed some studies that focus on how to apply the *as-a-service* approach to the different security domains involved in a cloud [21]. However, no real solution is available at the state of the art. Offering Security-as-a-Service is a complex problem, as security mechanisms are *embedded* into existing infrastructures and crosswise cover more offered services.

Analyzing the five characteristics and applying them to the Security-as-a-Service paradigm, we can outline that *on-demand self-service* means that customers should be able to activate security features without human intervention. This implies the need for automatic negotiation and agreement support. Furthermore, security features should be applied with a user-centric approach, i.e., not starting from what the provider offers, but from user needs.

Considering the *broad network access* characteristic, we should consider that security services have to be offered through an open access to the Internet, offering easy means to configure and to activate security features. Once again, the user-centric prospective of security becomes a requirement.

Moreover, due to the *resource pooling* characteristic, Security-as-a-Service should be applied in a dynamic way. The same service should be offered in different ways, if different security services are activated. Due to *rapid elasticity*, an already-offered service should be easily reconfigured to adapt to the activation of a new security service.

Last but not least, the *measured service* characteristic implies the capability to evaluate the security level actually granted to end users. As in clouds the measurement of the quality of services is an issue, in the context of security this problem grows up, due to the absence of stable and standard solutions to measure security quantitatively.

Security-as-a-Service is a need but not yet the solution. Cloud customers would like to have a simple way to add new security features to their existing services, but their implementation and deployment raise several issues as regards the intrusiveness of security in managed cloud infrastructures and the possibility to evaluate the security of these services with respect to their customers.

III. The SPECS Approach

The approach we propose is based on the notion of security parameters included in cloud SLAs, which would help end users and CSPs to have a *common language* to negotiate security requirements/offerings. The core idea in SPECS is that security management in SLA will enable the negotiation, the continuous monitoring and the enforcement of agreed security levels. This is the basis for offering Security-as-a-service [12].

An end user may negotiate his security features through the dedicated SPECS platform, which allows him to choose the right service to be invoked, by automatically enforcing the required security mechanisms and/or by applying any additional mechanisms in order to fulfill an agreed SLA.

From user perspective, the agreement on a SLA defines the way in which services are protected and accessed, whereas in-house system administrators use the agreed SLA to configure the way in which cloud services are offered. This approach was sometimes experimented in the past [17], [32].

A typical SLA life cycle can be characterized by three main phases: Negotiation, Monitoring and Enforcement.

In the *Negotiation* phase, the SLA is not fully defined, and the customer(s) and provider(s) conduct a negotiation process on requirements/services to find agreement on what the SLA should effectively offer. The customers evaluate the trade-off across service specifications (base services and options), and the corresponding performance and costs. Service Providers, instead, have to evaluate the services requested, matching to what can be actually granted and also need to evaluate the risks related to incorrect evaluation.

During the *Monitoring* phase, a signed SLA is checked for its actual degree of conformance or for penalties if in violation. Note that at the state of the art no systematic SLA-monitoring solutions exist to support users. This is typically a cumbersome manual process. From the Service Provider view, Monitoring implies two different activities: (a) verifying that the SLAs are respected via access to the underlying infrastructure that is inaccessible to end users, and (b) generating alerts before SLAs are broken, in order to activate remedial actions.

The final step of the SLA life cycle is the *Enforcement*, where the actions needed to respect the SLA (i.e., to keep a sustained QoSec) are effectively taken. This may imply the activation of software modules, the acquisition of resources (in the correct amount), but possibly even the dynamic reconfiguration of resources after an alert is generated.

The three phases are correlated: the Negotiation cannot be performed without taking in consideration how SLA can be granted, i.e., how the Enforcement will take place. Enforcement needs Monitoring in order to evaluate the real state of the solution before applying the predefined policies and procedures, while Monitoring needs the results of negotiation to know what to monitor and which alerts should be generated.

A Platform-as-a-Service supports both the development of new security-related services and the full SLA life cycle. Such SLA-based platform dedicated to security services will be able to integrate existing solutions inside a cloud Provider to configure the security services using internal tools. However, it can be also used as a third-party (i.e., as a broker) dedicated to secure cloud service brokering and to improve CSP offerings with additional services dedicated to security. The SPECS approach aims at solving the outlined cloud security open issues, offering tools to manage the security requested in SLAs as a basis to enable Security-as-a-Service. Moreover, such solutions can be offered by third parties (brokers), thus allowing end user monitoring functionalities, e.g., the possibility to notify about SLA violations due to cyber incidents.

IV. SPECS Scenarios and Services

SPECS software is expected to be distributed as a cloudware, which can be used to setup and instantiate the SPECS platform on local or cloud-based resources, in order to offer security services to end users.

SPECS Security-as-a-Service solution is based on the notion of systematically managing/reasoning about the security parameters defined on a CSP Service Level Agreements, which are explicitly agreed between cloud customer and CSP. Following such approach, SPECS should offer a set of services which manage the full life cycle of an SLA (i.e., negotiation, monitoring and enforcement). SPECS services should be able to guarantee the fulfilment of a minimum agreed SLA in order to provide a set of required security features.

The proposed SPECS PaaS faces the problem of a clear definition of the responsibilities in SLA. We have the following actors involved: (i) end users (which use SPECS and providers services), (ii) SPECS administrators (which manage the SPECS platform and offer SPECS services), (iii) CSPs. They may interact in different ways, according to two different interaction models, as illustrated in next subsections.

Consequently, the SPECS Platform offers two different classes of services that imply different responsibilities for the involved actors:

- *Improved Security*: SPECS providers sell security services and assume the responsibility of the SLA they offer. If a SLA is not fulfilled, then the responsibility is on the SPECS administrator (not to the CSP);
- *Brokering*: SPECS acts only as a secure-oriented broker, assuming no liability. In this case, SPECS services are offered by CSP and the platform only refers to the SLA offered by these CSPs.

In both cases the SPECS platform has the role of offering to its administrators the services needed to fully manage (i.e., monitor, negotiate and enforce) an SLA and the interactions among the different stages of its life cycle. The Core Services are the glue among the different modules that enable the full achievement of the Security-as-a-Service approach. They are of few different typologies (which may be refined during project execution), as described in Table I.

The Core SLA-based services are responsible for the full achievement of the SLAs. In fact, while different modules have the role of controlling an SLA evolution, the Core services have the role of maintaining the state of the SLA itself. These services aim also at helping end users and administrators to identify the responsibilities related to the kind of services being offered. Communication Services aim at enabling the interactions among the logical modules of the SPECS solution. As mentioned before, each stage of the SLA needs continuous interactions with other stages, thus implying that we need services that help such interactions, maintaining independence among modules. The final class of SPECS services, namely Utilities, is development-oriented, and is intended as a set of helper modules for developers customizing the offered solution.

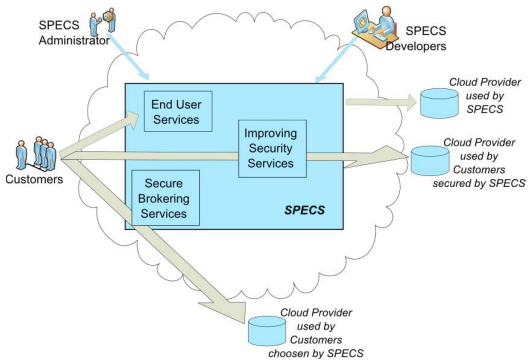

Fig. 1. SPECS *Third Party* Interaction model

A. SPECS Usage Scenarios

The SPECS PaaS has three possible usage scenarios. They imply different ways to use the platform and to run cloud applications and services:

- *Third Party Security Platform*: SPECS administrators run the Platform on resources acquired from public/private cloud and offer security services to cloud end Users. Such services enrich cloud services offered by independent CSP.
- *Hosted Platform*: SPECS runs on local resources and it is used to offer additional security services in order to sustain an agreed QoSec.
- *User Software*: cloud end users that use SPECS on their local resources to deploy required security features, additional to those offered by their CSPs.

1) Third Party Security Platform: The main scenario for the SPECS solution is illustrated in Fig. 1: the platform runs as an independent third party component, managed by a SPECS administrator, offering security services to end users over CSP services which are not aware of the SPECS intermediary actions. In this scenario, we envision a push architecture where an end-user invokes SPECS services to augment the security of its request before accessing any cloud service. In this way, the provided services are intended to improve existing security level of invoked cloud services (namely improved security services) by providing a kind of filter (secure broker) before accessing any cloud service that does not need to change its interfaces, APIs, ... For example, cloud users may negotiate the adoption of DoS protection services for a Web Server leased from a cloud Provider. The SPECS services will start an Intrusion Tolerant service (provided with internal security services) on the provider, in a transparent way.

The SPECS platform acts as a secure broker which adds security features over third party services. The PaaS offers to end users three class of services: internal services related to interactions between end users and SPECS platform, services

TABLE I. CORE SERVICES CATEGORIES

Service	Type Description	Example(s)
SLA-based	Services related to SLA management	SLA secure repository (to securely store agreed SLAs) and implement SLA query services (to retrieve information about agreed SLAs)
Communication	Services dedicated to interactions among different modules of the architecture	Services enabling the Monitoring module to generate alerts for the Enforcement module
Utility	Services which are not strictly needed, but can be of common help	Parsing services, format converters, etc

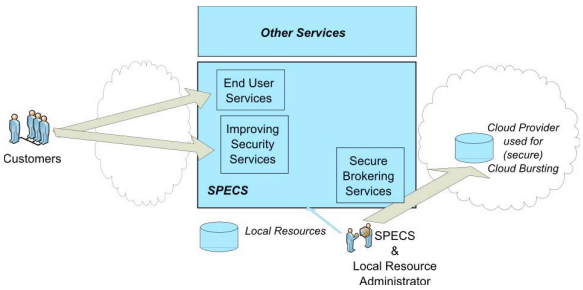

Fig. 2. SPECS *Hosted Platform* Interaction model

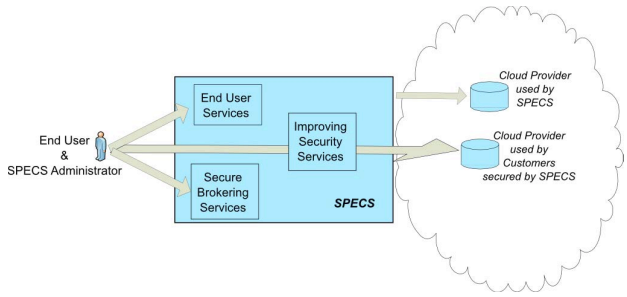

Fig. 3. SPECS *User Software* Interaction model

which enrich the security features of a target service and services which enable secure brokering. So, end users use SPECS in order to identify the best way to involve the target services, then invoke them directly. The PaaS consumes resources independently from the target services, which are directly managed by SPECS administrators.

2) Hosted Platform: In the second scenario, the SPECS platform can be adopted by a CSP in order to offer advanced security features over the services and resources being normally offered to end users. Fig. 2 shows this interaction model.

In such interaction scheme, the SPECS administrators are co-located within the CSP, and so they use the contributed platform in order to offer advanced security services to their users. The classes of services offered are similar to the above described ones, but the trust issues and the usage implications differ: the CSP can integrate SPECS services into their solutions or just use their alternative security services internally developed for the other services. End user security services, that in the first interaction model act as intermediaries among end users and providers, in this case are used only to enrich well-known features and can be customized to a single API and offering.

3) User Software: The last scenario assumes that the platform is dedicated to a single end user, which runs the platform to manage its own activities (Fig. 3). In this case, SPECS does not offer services to end users, but it is used by its administrator to manage security features over remote services. Internal services are almost useless (no need for authentication or offering something internally), while brokering and secure invocation are the main goals of the solution. In this case there are no dedicated components for credential management, because the end user is also the SPECS administrator.

Apart from these scenarios, the PaaS solution used within SPECS aims at reusing as much as possible existing software architectures in order to motivate the adoption of the framework. The project aims at offering an open-source, secure-oriented platform Security-as-a-Service that avoids the need for developers to fully create a new cloudware solution.

V. POSITIONING OF SPECS APPROACH

The cloud has become a computing/communication paradigm that seems to have the potential to change the way we consider systems and services. Unfortunately, as pointed out by ENISA the European Network and Information Security Agency in their report [8], the cloud also conveys serious security and privacy issues that nowadays represent major show-stoppers for its adoption, namely: loss of governance, vendor lock-in, isolation failure, compliance risks, management interface compromise, data protection, insecure/incomplete data deletion and malicious insiders.

Another well-known report on cloud computing security was developed by the Cloud Security Alliance [6], where, in analogy to ENISA report, cloud-specific risks are classified in thirteen domains divided into cloud architecture, governance and operational factors. Despite the amount of scientific papers on this topic (e.g., ACM published approximately 2000 articles in the period 2010-2012), the idea of providing security assurance as a mean to achieve cloud security and trust has prevailed. The recent efforts on specification of security statements in SLAs are a positive development and also the starting point for the SPECS project.

A. Specifying security parameters in Cloud SLA

The use of cloud SLAs has the potential to provide tangible benefits to CSPs. The specification of security parameters in SLAs forces a CSP to explicitly address security. The end users can also benefit from SLAs by understanding the costs and benefits associated with this new service model. On one hand, SLAs aim to provide service-based assurance. On the other, it is clear that SLAs are not intended to replace assurance mechanisms for security policy enforcement [20].

The importance of cloud SLAs has also been recognized by ENISA: the development of template contracts and service level agreements (SLA) is one of the areas to be addressed in the European cloud computing strategy. In a recent survey [15] ENISA highlights that many cloud customers often do not monitor security aspects of their contracted SLA on a continuous basis. This implies that customers are left unaware about important security aspects. The risk is that they find

out about failing security measures only after a security breach. The survey data show that while SLAs are often used, and availability is often addressed in these SLAs, security parameters are less well covered. Bernsmed [5] stated that a cloud SLA usually models the CSP security at the service level and is based on either a set of expert-driven security requirements (e.g., for compliance reasons) or some kind of preliminary threat analysis. The result is a collection of security statements (also called security provisions) in the form {security attribute, value} (e.g., {Backup Frequency, Daily} and {Encryption Key Size, 512 bits}) [7], [22], [34]. To be manageable, these security provisions must be organized into "categories" derived from a taxonomy, e.g., Savola [35] or the Cloud Security Alliance Cloud Controls Matrix [13]. This set of security provisions, now organized into taxonomic categories, will finally become a cloud SLA "template", to be used by CSPs to define their own SLAs. Cloud SLAs are usually stored in publicly-available and trusted repositories as the CSA Security, Trust & Assurance Registry (STAR [14]).

Apart from the challenges related with the creation of SLAs in real cloud deployments, the current paucity of techniques to quantitatively reason about them has proven to be part of the obstacles in using SLAs, as pointed out by Almorsy [1] and Luna [22], [23], [24].

B. Negotiation of security in cloud SLAs

The negotiation processes among software agents and/or humans is an active research topic, which involves a lot of different aspects, mainly focused on the goal of defining techniques to reach an agreement among different actors, which may have different goals and perspectives on the system. A negotiation can be classified based on the items being negotiated (single-attribute items or multiple-attribute items), the characteristics of the negotiators, the number of parties (one-to-one, one-to-many, or many-to-many), the negotiation protocol, the characteristics of information (completeness and symmetry), the negotiation period (continuous, one step, multiple stage), and other factors (openness, with penalty, ...).

The main focus of research activities is related to definition of techniques able to automate multi-tenants agreement models and protocols, in order to develop software agents able to take care of such processes without human intervention (see [3], [10], [16]). In the context of SLA negotiation, few results are actually available, even if tentative standard solutions exist, such as WS-Agreement Negotiation [2]. Protocols adopted in such a context are focused on adoption of Web Services. More recent results [21] propose REST-based protocols and APIs.

The adoption of security parameters in SLA is a novel problem. As outlined by Meland et al. in [27], state-of-the-art languages often are not able to fully represent security needs in SLAs. Many activities are ongoing on this topic, in many related projects (as A4CLoud and CUMULUS) and in independent research activities [27], [30].

The SPECS project aims at reusing as much as possible such existing solutions and proposals of other projects. To the best of our knowledge, there are only two previous works related with the idea of negotiating security in cloud SLAs. The first one was contributed by Rak [32], where authentication and authorization mechanisms are negotiated between users and CSPs via a SLA-based interface in the context of the mOSAIC project [31]. Hale [19] introduced SecAgreements, a framework for negotiating cloud security risks via (i) a SLA-based matchmaking algorithm and, (ii) a set of extensions proposed for the WS-Agreement protocol [2].

C. Enforcement of SLA

Currently SLA enforcement focuses on configuration of services after SLA agreement between service provider and end user. To our knowledge, there is no existing work trying to improve security following up the enforcement of an SLA. In general, SLA enforcement is performed in two ways: through active enforcement of the property value during service initiation at CSP (e.g., when creating or deploying a VM in the IaaS), or by monitoring and exception reporting.

Moreover, the approach of offering Security-as-a-Service, which is strictly related to the capacity to enforce security mechanisms in the SPECS perspective, is just at an explorative stage, as outlined by the CSA activities, which currently are mainly a classification of the offered services and a starting requirement analysis. No real implementations are available.

In the context of SPECS, we will assume CSA results as a basis for our enforcement services requirement and as a starting point for the services definition and implementation in the context of the SPECS project.

D. Monitoring, SLA and cloud

Monitoring solutions at cloud level differ as for the type of services offered. The most common solutions are related to the Infrastructure-as-a-Service level. In such cases, monitoring maps at resource and performance classes proposed above. Two monitoring approaches exist for this: adoption of standard monitoring solutions, typically used for physical hardware monitoring (as Ganglia [26], Sflow [37], Nagios [4]), re-adapted for the cloud environment, or native cloud monitoring services, often provided by CSPs (e.g., Amazon CloudWatch). Few cloud-oriented monitoring solutions have been proposed, as the one adopted in the mOSAIC project, which aims at mixing all the monitoring layers [33].

The problem is worse as for Security Monitoring. This can be done: (i) by CSPs, through their software stack, using data available at the CPS side, or (ii) by users, using data from the CSPs or collected through "sensors" in the client environment. As specified by Anton Chuvakin in "Security Monitoring of Public Cloud Assets" [11], cloud security monitoring is almost an afterthought for most organizations and as cloud usage expands. Security monitoring must be deployed across public clouds, private clouds and traditional infrastructures. Users, and not providers, own that responsibility.

Security monitoring, moreover, include aspects related to continuous monitoring for detection of intrusion and malicious attacks. In this case, the problem is related to apply real-time elaboration on data collected by resource monitoring, as discussed above. Even if such techniques are commonly adopted in Intrusion Detection/Tolerance Systems, few results are available in the cloud environment [29], [36]. Cheng et al. [9] present a distributed IDS architecture for clouds. They propose to separate the IDS for each cloud customer from the actual component being monitored. Similarly, Chi-Chun et al. [25] propose a federation defense approach, in which the IDSs are deployed in each cloud computing region. However, currently there is no real implementation of the proposed approach. Yassin et al. [38] propose an intrusion detection framework based on cloud services for commercial solutions and research communities. However, it only sniffs packets at network level. As preliminary work for SPECS, Rak et al.

[17] propose that IaaS is enriched with ad-hoc solutions for protecting the delivered resources against a set of security attacks. Moreover, Ficco et al. [18] proposed a mOSAIC-based framework for distributed intrusion detection in clouds.

VI. Conclusions

In this paper we introduced the SPECS concepts and platform architecture, as a solution for offering Security as-a-service using a SLA-based approach. This paper stems from the SPECS project startup and outlines the core architecture of the proposed solution, but it takes advantages of results from previous work and projects.

The main benefit of using the SPECS approach will be the chance to offer security assurance to cloud end users in terms of cloud services, by managing the life-cycle of agreed security parameters contained in cloud SLAs.

Beyond the concept implementation, in the near future we aim at exploiting the details of the core services and of the functionalities offered for each of the three aspects of the SLA life cycle. Moreover, we will propose simple case studies to apply the platform on real case studies.

Acknowledgment

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

References

[1] Almorsy M., et.al. Collaboration-Based Cloud Computing Security Management Framework. In Proc. of IEEE Intl Conference on Cloud Computing, 364-371, 2011

[2] Andrieux K., et al. Web Services Agreement Specification (WS-Agreement). Technical Report TR-WSAgreement-2007, Open Grid Forum, 2007.

[3] Aversa, R., et al."User centric service level management in mosaic application". LNCS 7156, 106-115, 2012.

[4] Barth, W. "Nagios System and Netwok Monitoring". William Pollock Editor, 2008.

[5] Bernsmed K., et al. Security SLAs for Federated Cloud Services. In Proc. of IEEE Availability, Reliability and Security, 202-209, 2011.

[6] Brunette, G. and Mogull, R. Security Guidance for critical areas of focus in Cloud Computing. Technical Report Cloud Security Alliance. 2009

[7] Casola V., et al. A SLA evaluation methodology in Service Oriented Architectures. In Quality of Protection, Springer Advances in Information Security 23, 119-130. 2006.

[8] Catteddu, D. and Hogben, G. Cloud Computing Risk Assessment.Technical Report. European Network and Information Security Agency (ENISA). 2009.

[9] Cheng F. and Meinel . "Intrusion Detection in the Cloud". In Proc. of the IEEE Int. Conf. on Dependable, Autonomic and Secure Computing, 729-734, 2009.

[10] Chhetri M. et al. "A coordinated architecture for the agent-based service level agreement negotiation of web service compositions". In The Australian Software Engineering Conference. Springer, 2006.

[11] Chuvakin, A.. "Security Monitoring of Public Cloud Assets". Gartner, 2012. Available at http://blogs.gartner.com/anton-chuvakin/2012/01/09/cloud-security-monitoring/

[12] Cloud Security Alliance: "Security-as-a-Service". Available at https://cloudsecurityalliance.org/research/secaas/#_overview, 2012

[13] Cloud Security Alliance. The Consensus Assessments Initiative Questionnaire". Available https://cloudsecurityalliance.org/research/cai/, 2011.

[14] Cloud Security Alliance. The Security, Trust & Assurance Registry (STAR). Available at https://cloudsecurityalliance.org/star/, 2011.

[15] Dekker M. and Hogben G. Survey and analysis of security parameters in cloud SLAs across the European public sector. Technical Report TR-2011-12-19, European Network and Information Security Agency, 2011

[16] Emeakaroha, V.C., et al. Low level metrics to high level slas - lom2his framework: Bridging the gap between monitored metrics and sla parameters in cloud environments. Information Systems Journal, 2:48-54, 2010.

[17] Ficco, M. and Rak, M. "Intrusion tolerance as a service: A SLA-based solution". In Proc. of the 2nd Int. Conf. on Cloud Computing and Services Science, 375-384. IEEE CS Press, 2012.

[18] Ficco, M., et al. "mOSAIC-Based Intrusion Detection Framework for Cloud Computing". LNCS 7566, 628-644, 2012.

[19] Hale M.L. and Gamble R. SecAgreement: Advancing Security Risk Calculations in Cloud Services. In Proc. of the IEEE World Congress on Services, pages 133-140, 2012.

[20] Henning R. Security service level agreements: quantifiable security for the enterprise? Proc. of ACM Workshop on New security paradigms, 54-60, 1999.

[21] Kabert, R., Katsaros, G., and Wang T. "A RESTful implementation of the WS-agreement specification". In Proceedings of the Second International Workshop on RESTful Design, 67-72, 2011

[22] Luna J., et al. A Security Metrics Framework for the Cloud. In Proc. of Security and Cryptography, 245-250, 2011.

[23] Luna J., et al. Quantitative Assessment of Cloud Security Level Agreements: A Case Study. In Proc. of Security and Cryptography, 2012.

[24] Luna J., Langenberg R. and Suri N., Benchmarking Cloud Security Level Agreements Using Quantitative Policy Trees In Proc. of the ACM Cloud Computing Security Workshop. 2012.

[25] Lo C.C., Huang C.C., and Ku J. "A Cooperative Intrusion Detection System Framework for Cloud Computing Networks". In Proc. of the 39th Int. Conf. on Parallel Processing, 280-284. IEEE CS Press, 2010.

[26] Massie M., Li B., Nicholes B., Vuksan V., Alexander R., Buchbinder J., Costa F., Dean A., Josephsen D., Phaal P., Pocock D. "Monitoring with Ganglia Tracking Dynamic Host and Application Metrics at Scale". OReilly Media, 2012.

[27] Meland, P.H., et al. "Expressing cloud security requirements in deontic contract languages", CLOSER 2012 - Proceedings of the 2nd International Conference on Cloud Computing and Services Science, 638-646, 2012.

[28] Mell P. and Grance T. The NIST Definition of Cloud Computing. Technical Report. National Institute of Standards and Technology (NIST). 2009.

[29] Park M.W. and Eom J.H. "Multi-level Intrusion Detection System and Log Management in Cloud Computing". In Proc. of the 13th Int. Conf. on Advanced Communication Technology, 552-555. IEEE CS Press. 2011.

[30] Pearson, S., Tsiavos, P. "From creative commons to smart notices: Designing user centric consent management systems for the cloud". CLOSER 2012 - Proceedings of the 2nd International Conference on Cloud Computing and Services Science, 647-660, 2012.

[31] Petcu, D., et al. "Experiences in building a mOSAIC of clouds", Journal of Cloud Computing: Advances, Systems and Applications 2 (1), 2013.

[32] Rak M., et. al. A SLA-based interface for security management in cloud and GRID integrations. In Proc. of the IEEE Intl. Conf. on Information Assurance and Security, 378-383, 2011.

[33] Rak, M. et al. "Cloud application monitoring: The mosaic approach". In Proceedings - 2011 3rd IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2011, 758-763, 2011.

[34] Samani R., et al. Common Assurance Maturity Model: Scoring Model. Available http://commonassurance. com/, 2011.

[35] Savola R., et al. Towards Wider Cloud Service Applicability by Security, Privacy and Trust Measurements. In Proc. of IEEE Application of Information and Communication Technologies, 1-6, 2010.

[36] Schulter, K.. "Intrusion Detection for Grid and Cloud Computing". In IEEE IT Professional Journal, July 2010.

[37] Wang, M., et al. "sFlow: towards resource-efficient and agile service federation in service overlay networks," Proceedings. 24th International Conference on Distributed Computing Systems, 628- 635, 2004.

[38] Yassin, W., et al. "A Cloud-based Intrusion Detection Service framework". In Proc. of the Int, Conf, on Communication, Networking & Broadcasting, 213-218. IEEE CS Press. 2012.