05/03/2017 IEEE Xplore Full-Text HTML : A Security SLA-driven Methodology to Set-Up Security Capabilities on Top of Cloud Services

IEEE.org | IEEE *Xplore* Digital Library | IEEE-SA | IEEE Spectrum | More Sites Cart (0) | Create Account | Personal Sign In

Access provided by:
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
Sign Out

BROWSE MY SETTINGS GET HELP WHAT CAN I ACCESS?

QUICK PREVIEW Abstract Authors Figures Multimedia References Cited By Keywords

A Security SLA-driven Methodology to Set-Up Security Capabilities on Top of Cloud Services

The extensive use of cloud services by both individual users and organizations induces several security risks. The risk perception is higher when Cloud Service Providers (CSPs) do not clearly state their security policies and/or when such policies do not directly match user-defined requirements. Security-oriented Service Level Agreements (Security SLAs) represent a fundamental means to encourage the adoption of cloud services in contexts where security is mandatory. Nevertheless, despite the number of existing initiatives aimed at formalizing Security SLAs and at representing security guarantees by taking into account both customers' and providers' perspectives, they are far from being commonly adopted in practice by CSPs, due to the difficulty in automatically enforcing and monitoring the security capabilities agreed with customers. In this paper we illustrate, through a case study, a methodology to set-up a catalogue of security capabilities that can be offered as-a-service, on top of which specific guarantees can be specified through a Security SLA. Such a methodology, which explicitly takes into account the constraints behind the definition of formal guarantees related to security, is meant to serve as a guideline for providers willing to offer for their services specific security features that can be monitored and assessed by customers during operation.

This paper appears in: 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS), Issue Date: 6-8 July 2016, Written by: Valentina Casola; Alessandra de Benedictis; lina Era_cu; Massimiliano Rak; Umberto Villano

SECTION I
INTRODUCTION

Cloud services represent a big portion of the present IT industry. With the wide adoption of the cloud computing paradigm, more and more organizations and individual customers rely upon cloud services to carry out their business. However, reliance on services provided by third-parties, whose use is possibly shared among different customers, carries several concerns related to decreased control over personal data and sensitive information. This may have a negative influence on CSPs' business goals and represents, at present, the stronger limitation in the adoption of cloud services for those customers that need security guarantees. The lack of providers' transparency with respect to security and the resulting decreased confidence of customers in offered services is partly due to the fact that CSPs and customers typically look at security from very different perspectives. Unfortunately, customer-defined requirements often do not directly match the information supplied by CSPs regarding the security levels associated with their services, as providers frequently express guarantees through a technical, low-level language, which is hard to understand for non-technical operators.

Extensive research activities have been recently carried out, in the context of both academical research and industry and government-driven initiatives, on the definition of Security SLAs and on their application to cloud environments [1]. Regarding the identification of security parameters, several guidelines and international standardization initiatives exist that aim at defining a shared catalogue of *security controls* related to both technical and non technical aspects of security (i.e., the ISO27002 standard [2], the NIST Security Control Framework [3], and the Cloud Control Matrix-CCM-from Cloud Security Alliance [4]). Such security control frameworks are mainly intended to be used by organizations to assess the level of security of their services by specifying what is the set of security controls (i.e., the security capabilities) that are actually enforced. Despite the strong interest in security and the existing efforts towards standardization, Security SLAs are far from being commonly adopted by CSPs, which at state of art mainly report performance-related parameters in their SLAs. Customers can only accept delivered services as they are, without the possibility of negotiating and, above all, of monitoring the level of security of the services they acquire.

The work reported in this paper is partly related to the activities carried out in two EU projects, namely SPECS[1] and MUSA [2], whose objectives are respectively to provide a platform-as-a-service to develop SLA-based cloud security services and to promote security-by-design in multi-cloud application contexts through the adoption of Security SLAs. In particular, in this paper, we propose and illustrate a practical methodology to map customer-defined requirements to providers' offered capabilities based on existing security control frameworks' guidelines. Such a methodology is meant to serve as a guideline for providers willing to offer security features on top of their services while also providing some guarantees related to those features. The novelty of our contribution with respect to the current state of the art consists in explicitly taking into account the constraints set by the definition of formal guarantees related to security, namely the need for identifying proper metrics and related Service Level Objectives (SLOs) to enforce and monitor the fulfillment of related requirements during system operation.

This paper is structured as follows. Section II presents some relevant background about Security SLAs and existing initiatives aimed at defining security controls and security SLOs, and introduces our motivation and the context of our contribution. Section III briefly illustrates the SPECS approach to cloud security through SLA management with particular focus on the construction of secure supply chains, while Section IV reports on the methodology proposed to enable their automatic management. Section V describes an example of application of the proposed methodology to a web container service and, finally, Section VI draws our conclusions.

SECTION II
Motivation and Background

The concept of a Security Service Level Agreement to specify the requirements of security services for an enterprise was first proposed by Henning [5] in 1999, and has been widely adopted since then to identify a contractual agreement between a service provider and a service customer which explicitly contains guarantee terms related to security properties. In 2011, ENISA published a report analyzing the use of security parameters in cloud SLAs (mostly focused on the EC public sector) [6], which put in evidence that, although security was considered by most respondents as a top concern and SLAs were actually often adopted by CSPs, they typically addressed only availability and other performance-related parameters. Moreover, the survey outlined that the tools (in terms of regular reports on measurements and incidents) provided to customers to let them monitor the security of acquired services were generally inadequate. The subsequent report by ENISA [7] built on previous work and aimed at giving guidance to customers on how to continuously monitor the security service levels and governance of outsourced cloud services. This was achieved through the reporting and alerting of key measurable parameters, as well as through a clear understanding of how to manage the customers responsibilities for security.

In order to promote the adoption of security best practices and aid the process of security management for enterprises, several standard initiatives have been proposed in the last years, aimed at defining shared catalogues of security controls. ENISA's Information Assurance Framework [8], released in 2009 and based on ISO *27001/2* standards (cf. [9], [2]) and on industry best practice requirements, was designed to help organizations assess the risk related to the adoption of cloud services and to compare different offers with respect to security properties. Similarly, NIST released its Security Control Framework (NIST sp800-53 [3]), which provides a three level-hierarchy of security controls addressing areas such as mobile and cloud computing, applications security, trustworthiness, assurance, and resiliency of information systems. Finally, the Cloud Control Matrix [4] released by the Cloud Security Alliance provides fundamental security principles to guide cloud vendors and to assist prospective cloud customers in assessing the overall security risk of a CSP. It is based on NIST sp800-53 and on other industry-accepted security standards and is structured in 16 domains.

Despite the effort spent in investigating the adoption of Security SLAs in the cloud and in the formalization of security capabilities by means of security control frameworks, the situation has not changed much since the publication of the ENISA report, as the most popular CSPs (such as, for example, Amazon and Google) still do not offer SLAs including security-related guarantee terms but only report the security features their services come with, by possibly giving some technical details on their implementation. The customer, therefore, has to accept the service *as-it-is*, and is typically not provided with any assurance related to the level of security associated with a service, nor with any means to monitor the actual fulfillment of the requirements declared by providers. With respect to the above discussed issues, our work aims at enabling the actual adoption of Security SLAs in cloud, by building a solution for improving the user-centric negotiation of security level objectives, automating the enforcement of related security capabilities and monitoring the associated security metrics. This is actually the approach proposed in the SPECS project, which promotes the development of a framework for the management of the whole Security SLA life-cycle, but it is also the basis for enabling cloud providers to clearly state their security offer with respect to customers' needs. In the next sections, we briefly illustrate the SPECS approach with particular focus on the enforcement of security *as-a-Service* according to an agreed Security SLA, and discuss the methodology adopted to enable its automation.

SECTION III
Enforcing Security-as-a-Service: The SPECS Approach

As said, the SPECS project aims at designing and implementing a framework for the management of the whole SLA life cycle, intended to build applications (the SPECS Applications) devoted to offering services to SPECS Customers, whose security features are stated in and granted by a Security SLA [10]–[11].

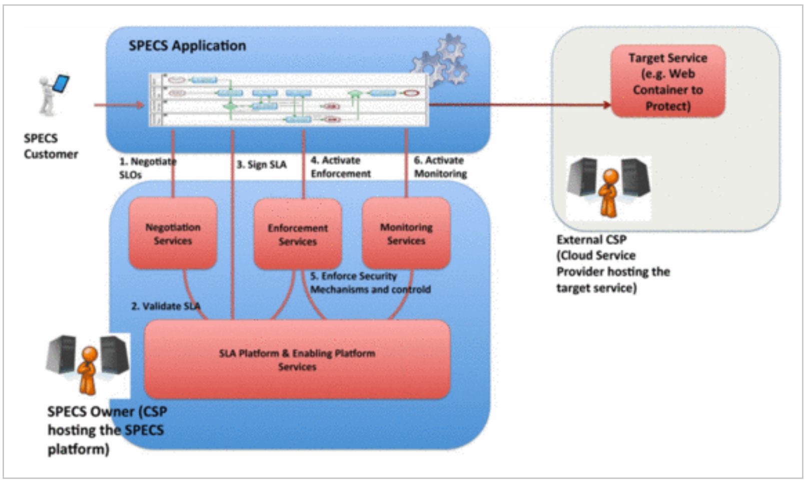

SPECS Application
SPECS Customer
1. Negotiate SLOs
3. Sign SLA
4. Activate Enforcement
6. Activate Monitoring
Negotiation Services
Enforcement Services
Monitoring Services
5. Enforce Security Mechanisms and controld
2. Validate SLA
SLA Platform & Enabling Platform Services
SPECS Owner (CSP hosting the SPECS platform)
Target Service (e.g. Web Container to Protect)
External CSP (Cloud Service Provider hosting the target service)

Fig. 1. The SPECS approach

As illustrated in Figure 1, a SPECS Application orchestrates the SPECS Core services dedicated to Negotiation, Enforcement and Monitoring respectively, in order to provide the desired service (referred to as the Target Services in the picture) to the SPECS Customer. Security-related Service Level Objectives (SLOs) are negotiated (Step 1) based on the SPECS Customer's requirements. A set of compliant offers is identified with the help of an interoperability layer (composed of the SPECS SLA Platform services), which is also responsible of their validation (e.g., to verify their actual feasibility based on the current system configuration) (Step 2). The agreed terms are included in a Security SLA that is signed by the SPECS Customer and the SPECS Owner (Step 3). Afterwards, the agreement is implemented through the Enforcement services, which acquire resources from External CSPs and activate proper components that provide, in an *as-a-Service* fashion, the security capabilities needed to fulfill the SLOs included in the signed Security SLA (Steps 4 and 5). At the same time, proper monitoring components are configured and activated (Step 6).

The methodology proposed in this paper and discussed in Section IV is strictly connected to the above discussed flow, since it aims at giving a guidance to build a catalogue of security services whose associated security capabilities can be negotiated and monitored by customers. The application of this methodology is the basis to enable the automatic enforcement of Security-as-a-Service, which has been extensively discussed in [12], and is a fundamental step toward fostering the actual adoption of Security SLAs for cloud service.

SECTION IV
Enabling the Automatic Enforcement of Security

Several approaches exist in the literature that aim at representing user-defined security requirements within Security SLA terms. Nevertheless, while the existing work mainly focuses on the translation of user-defined requirements to security controls defined in standard control frameworks and on the representation of security controls within machine-readable SLAs, they do not take into consideration the actual constraints behind this translation. Indeed, the representation of a security control in form of an SLA term (i.e., a security SLO) is only possible if some monitorable metrics exist that can be associated to the security controls, so that the desired level of security can be actually checked and proper countermeasures or penalties can be applied if needed.

Based on the above consideration, we propose a simple concrete methodology that can serve as a guideline for those providers who want to enrich their commercial offer by giving to their customers the opportunity to choose the security characteristics to apply to their conventional services and have them granted by a formal Security SLA. As illustrated in the previous section, we adopt this methodology in the context of the SPECS project to set-up and make available a catalogue of security services that can be activated on demand to secure services provided by External CSPs, whose security requirements have been negotiated by the customer. We consider the following steps (cf. Figure 2):

1. **Step 1.** *Identification of the security features to offer on a target cloud service*. The provider establishes what it is able and willing to provide in terms of security features (e.g., resilience to attacks, end-to-end encryption, etc.) on top of its services.
2. **Step 2.** *Identification of the related security controls*. With regards to the security features defined in the previous step, the reference *security control frameworks* are analyzed to identify the security controls that may be actually enforced. These controls are collected together to form one or more *security capabilities*, which are implemented through proper software/hardware tools and components.
3. **Step 3.** *Identification of monitorable metrics and/or enforceable parameters*. For each capability defined at the previous step, and possibly for each control of the capability, one or more monitorable metrics and enforceable parameters are defined. While metrics can be actually checked during system operation through proper associated monitoring services/systems, enforceable parameters represent configuration values that can be dynamically set to fulfill a specific control. Both of them must be verifiable, in the sense that it must be possible, in any moment, to check the value they assume.
4. **Step 4.** *Identification of admissible SLOs*. For each monitorable metric/enforceable parameter, the range of admissible values must be defined in order to set some constraints on the SLOs that can be requested by a customer.

Security capabilities and security controls identified at steps 1 and 2, as well as metrics and parameters determined at step 3 can be used to build a negotiation framework through which a customer can express his requirements and obtain a list of compliant offers to choose from. It is of fundamental importance in this context to clearly understand the role of the metrics and parameters identified in step 3 during negotiation and, later, during enforcement. In the negotiation phase, a customer submits his security requirements in a specific format depending on his security skills and in general on the application he is interfacing with. We assume such requirements are someway translated to the security capabilities to enforce (translation is up to the application and is out of the scope of our discussion). It is worth outlining that, during negotiation, the customer may either ask simply for the enforcement of a specific capability or go through the details of security controls and ask for guarantees on a specific level of service associated with a security control (by defining desired SLOs). While in the former case the enforcement of security implies an automatic activation of a security capability with predetermined characteristics, in the latter it may require a proper tuning and configuration of an available security capability to meet the SLOs. To enable the second possibility, which is far more interesting in the case of security-expert customers, they must be informed of the acceptable levels of services that they may request (and that can be enforced and guaranteed) related to a selected security control. Metrics and enforceable parameters associated to security controls are used to actually verify/ensure that the selected level of service objectives are met. In our work, we actually expose metrics and enforceable parameters to customers instead of service levels, therefore possible metrics' and parameters' values defined in step 4 of our methodology can be chosen by customers as desired service level objectives. During enforcement, all these metrics and parameters are used to properly configure the security capabilities.

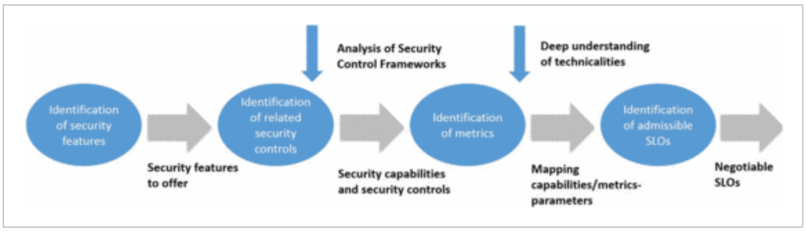

Analysis of Security
Control Frameworks
Deep understanding
of technicalities
Identification of security features
Security features to offer
Identification of related security controls
Security capabilities and security controls
Identification of metrics
Mapping capabilities/metrics-parameters
Identification of admissible SLOs
Negotiable SLOs

Fig. 2. Methodology overview

SECTION V
A Case Study

In order to illustrate the introduced approach, in this section we discuss a case study involving the provisioning of a secure web container service. Normally, one would acquire a Virtual Machine from an IaaS provider and proceed with the installation and configuration of a web container (e.g., Tomcat) on top of it, without having the possibility to explicitly negotiate and obtain specific security features. In our case study, the SPECS framework provides some additional security features to enhance web containers' protection, and a customer is enabled to negotiate and acquire them automatically. In Section V-A, we illustrate the process followed to set-up a negotiable capability that can be automatically identified based on requirements expressed by customers, and then enforced with an *as-a-Service* deployment model and continuously monitored. The described process is fundamental to enable the automatic negotiation of security features, as shown in Section V-B, where we report an example of negotiation with the SPECS Customer related to the acquisition of a *secure* web container service.

A. Set-Up of Security Capabilities

According to **Step 1 of methodology**, the first thing to do is the identification of the security features to offer on top of the web container service. In this case study, we consider to improve the *resiliency* of the web container service by ensuring a certain level of redundancy, through the activation and management of aligned web container replicas (along with a load balancer). Moreover, in order to increase the attack surface presented to a generic attacker, we also manage to use different web container technologies on the different replicas, subject to different vulnerabilities and threats.

Once the security features to offer have been identified, the associated security controls can be determined by analyzing the reference control frameworks (Step 2 of methodology). This task led us to identify the mapping reported in Table I, which lists control families, associated security controls and control enhancements related to the above discussed security features, tailored to the NIST Security Control Framework. The listed controls represent the *Web Resiliency* SPECS capability[3].

According to **Step 3 of methodology**, monitorable metrics and/or enforceable parameters must be identified for the *Web Resiliency* capability. In our case study, we considered the two following metrics:

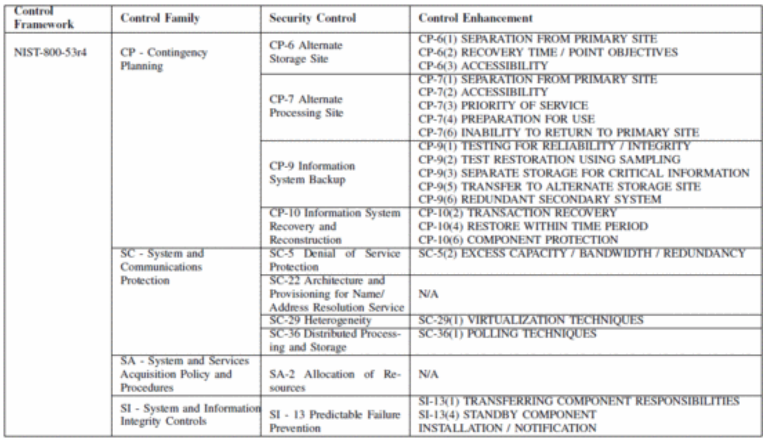
- *Level of Redundancy* is expressed as the number of replicas of the Web Container which are set-up and kept active throughout the service operation to ensure redundancy;
- *Level of Diversity* is represented by the number of different software and/or hardware versions of the Web Container service which are set-up and kept active throughout the service operation to increase the protection from attacks and vulnerabilities exploits.

Note that Level of Diversity (LoD) and Level of Redundancy (LoR) are both metrics, which can be easily checked by verifying the number of active replicas, and enforcement parameters, used to actually configure the service in order to achieve the desired resiliency. Moreover, note that LoD and LoR are correlated, in that it must be: $LoD \leq LoR$. Therefore, in the negotiation process, the customer must specify the desired values for these two metrics by taking into account this constraint.

The list of possible values associated with the above discussed metrics, defined in **Step 4 of methodology**, is reported in Table II. The admissible metric values are exposed to customers during negotiation for the selection of SLOs and are used during enforcement to configure the security capabilities accordingly. To the best of our knowledge, this approach has never been taken in consideration for negotiation and enforcement purposes.

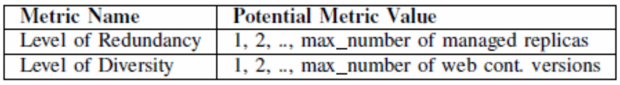
Finally, Table III summarizes the mapping between security controls identified for the *Web Resiliency* capability and respective metrics.

Note that, in general, this mapping may be very challenging as it requires deep understanding of the security standards and specific technicalities related to select security capabilities. However, we think that retrieving security controls for specific metrics is important if a customer wants to learn more about the security features he asks for and that are being offered. Furthermore, a mapping such as the one reported ensures that each specified control family/security control is enforced by a concrete service configuration through a specific configuration of the security capabilities available on top of the service itself. Depending on customers' security skills, the configuration of such capabilities may be even tuned ad hoc (possibly at an additional cost). Moreover, for some security controls, specific metrics are available to monitor the actual fulfillment of requirements. In the following subsection, we discuss different scenarios of interaction with the SPECS Customer that found on the set-up process here described.



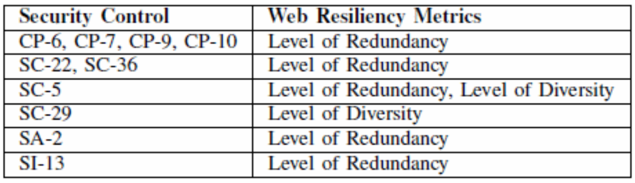
Control Framework	Control Family	Security Control	Control Enhancement
NIST-800-53r4	CP - Contingency Planning	CP-6 Alternate Storage Site	CP-6(1) SEPARATION FROM PRIMARY SITE CP-6(2) RECOVERY TIME / POINT OBJECTIVES CP-6(3) ACCESSIBILITY
		CP-7 Alternate Processing Site	CP-7(1) SEPARATION FROM PRIMARY SITE CP-7(2) ACCESSIBILITY CP-7(3) PRIORITY OF SERVICE CP-7(4) PREPARATION FOR USE CP-7(6) INABILITY TO RETURN TO PRIMARY SITE
		CP-9 Information System Backup	CP-9(1) TESTING FOR RELIABILITY / INTEGRITY CP-9(2) TEST RESTORATION USING SAMPLING CP-9(3) SEPARATE STORAGE FOR CRITICAL INFORMATION CP-9(5) TRANSFER TO ALTERNATE STORAGE SITE CP-9(6) REDUNDANT SECONDARY SYSTEM
		CP-10 Information System Recovery and Reconstruction	CP-10(2) TRANSACTION RECOVERY CP-10(4) RESTORE WITHIN TIME PERIOD CP-10(6) COMPONENT PROTECTION
	SC - System and Communications Protection	SC-5 Denial of Service Protection	SC-5(2) EXCESS CAPACITY / BANDWIDTH / REDUNDANCY
		SC-22 Architecture and Provisioning for Name/ Address Resolution Service	N/A
		SC-29 Heterogeneity	SC-29(1) VIRTUALIZATION TECHNIQUES
		SC-36 Distributed Processing and Storage	SC-36(1) POLLING TECHNIQUES
	SA - System and Services Acquisition Policy and Procedures	SA-2 Allocation of Resources	N/A
	SI - System and Information Integrity Controls	SI - 13 Predictable Failure Prevention	SI-13(1) TRANSFERRING COMPONENT RESPONSIBILITIES SI-13(4) STANDBY COMPONENT INSTALLATION / NOTIFICATION

Table I. Security controls related to the web resiliency capability



Metric Name	Potential Metric Value
Level of Redundancy	1, 2, .., max_number of managed replicas
Level of Diversity	1, 2, .., max_number of web cont. versions

Table II. Web resiliency metrics/parameters and potential values



Security Control	Web Resiliency Metrics
CP-6, CP-7, CP-9, CP-10	Level of Redundancy
SC-22, SC-36	Level of Redundancy
SC-5	Level of Redundancy, Level of Diversity
SC-29	Level of Diversity
SA-2	Level of Redundancy
SI-13	Level of Redundancy

Table III. Mapping between security controls and web resiliency metrics

B. Negotiating a Secure Web Container Service

A SPECS Customer represented by a web developer aims at acquiring a reliable web container. To obtain the web container with the desired features, the SPECS Customer accesses the SPECS Application and specifies his requirements by means of a wizard which enables him to navigate and select a set of capabilities and, possibly, of desired security controls. We discuss two scenarios: in the former, the SPECS Customer is not an expert in security field, therefore he is not aware of the best practices and of how to protect his web applications from malicious attacks, but he is aware of the technologies that may be involved. In the latter, the SPECS Customer is expert in security and is able to navigate the controls and choose desired values for respective metrics and parameters.

Scenario 1

A non-expert SPECS Customer accesses the wizard offered by the SPECS Application and is prompted with a description of the available capability, namely the *Web Resiliency* capability, whose set-up has been illustrated previously. The SPECS Customer selects the capability, and the SPECS Negotiation returns to the SPECS Customer a list of different (pre-built) offers, ordered based on the level of security they are able to provide. Each offer actually corresponds to a different enhanced supply chain, namely to a different configuration for the components offering the desired capabilities in combination with the web container service delivered by the CSP. The SPECS Customer chooses the offer labeled as the most secure one, and signs an SLA containing the default metrics/parameters (e.g., level of diversity=1 and level of redundancy=2). SPECS acquires a set of Virtual Machines from an IaaS CSP on behalf of the SPECS Customer (registered on SPECS), and sets-up and activates the components devoted to implement the Web resiliency capability.

Scenario 2

A SPECS Customer expert in security accesses the wizard offered by the SPECS Application and is prompted with the description of the *Web Resiliency* capability. Upon selection of this capability, the SPECS Customer is prompted with the list of associated security controls, as illustrated in Table I. The SPECS Customer is interested in the *System and Communications Protection (SC)* category, and in particular in the SC-5 (Denial of Service) control. The wizard returns the metrics/parameters associated to this control (cf. Table III) with respective admissible values (cf. Table II). The SPECS Customer selects the following: level of diversity=3 and level of redundancy=4. The SPECS Negotiation module identifies the components able to cover the selected controls and returns to the SPECS Customer a list of different configurations for such components. Each configuration represents a different supply chain characterized by its level of security and its cost. The SPECS Customer chooses his preferred configuration and signs an SLA containing the agreed metrics/parameters. SPECS acquires the needed resources on behalf of the SPECS Customer and sets-up and activates the needed security components.

SECTION VI
CONCLUSIONS

In this paper we have illustrated a methodology to map customer-defined requirements to providers' offered capabilities based on existing security control frameworks' guidelines. This work is part of a wider activity carried out in the H2020 MUSA Project and in the EC FP7 SPECS Project, aimed at enabling the actual adoption of Security SLAs in the cloud, by building a solution for improving the user-centric negotiation of security level objectives, automating the enforcement of related security capabilities and monitoring the associated security metrics. With respect to the existing literature that also focuses on building Security SLAs taking into account both customers' requirements and providers' offers based on control frameworks, we put more emphasis on the constraints that are behind the definition of formal guarantees related to security, mainly represented by the need for identifying proper enforceable security level objectives and monitorable metrics to respectively enforce and monitor the fulfillment of agreed requirements. To support our methodology, we also provided an example illustrating its application to set-up a concrete security capability that can be negotiated, enforced and monitored automatically through the SPECS solution.

Acknowledgment

This research is partially supported by the grant FP7-ICT-2013-11-610795 (SPECS) and H2020-ICT-07-2014-644429 (MUSA).

FOOTNOTES

[1] http.//www.specs-project.cu

[2] http://www.usa-project.eu

[3] Note that the SPECS project actually envisions a wider set of security mechanisms and controls, applicable to different types of services, but we only mention this for brevity reasons.

REFERENCES

1. V. Casola, A. De Benedictis and M. Rak and M. Felici and C. Fernmdez-Gago Eds
"On the adoption of security SLAs in the cloud"
Accountability and Security in the Cloud, ser. Lecture Notes in Computer Science, vol. 8937, pp. 45-62, 2015, Springer International Publishing
Show Context

2. International Organization for Standardization
2013
Show Context

3. National Institute of Standards and Technology
2013
Show Context

4. "Cloud Control Matrix v3.0"
Cloud Security Alliance, [online] Available : https://cloudsecurityalliance.org/download/cloud-controls-matrix-v3/
Show Context

5. R. R. Henning
"Security service level agreements: Quantifiable security for the enterprise?"
pp. 54-60, 2000, New York, NY, USA, [online] Available : http://doi.acm.org/10. 1145/335169.335194
Show Context

6. M. Dekker and G. Hogben
"Survey and Analysis of Security Parameters in Cloud SLAs Across the European Public Sector"
ENISA, Tech. Rep., December, 2011
Show Context

7. "Procure Secure. A guide to monitoring of security service levels in cloud contracts"
ENISA, April, 2012
Show Context

8. "Information Assurance Framework"

November, 2009

Show Context

9. International Organization for Standardization

2013

Show Context

10. V. Casola, A. De Benedictis, M. Rak and U. Villano

“Preliminary design of a platform-as-a-service to provide security in cloud”

pp. 752-757, April 3-5, 2014., 2014, Barcelona, Spain

Show Context

11. M. Rak, N. Suri, J. Luna, D. Petcu, V. Casola and U. Villano

“Security as a service using an sla-based approach via specs”

vol. 2, pp. 1-6, 2013

Show Context

12. V. Casola, A. De Benedictis, M. Rak, J. Modic and M. Erascu

“Automatically enforcing security slas in the cloud”

IEEE Transactions on Services Computing, no. 99, pp. 1-1, 2016

Show Context

AUTHORS

Valentina Casola

No Bio Available

Alessandra De Benedictis

No Bio Available

Mădălina Eraşcu

No Bio Available

Massimiliano Rak

No Bio Available

Umberto Villano

No Bio Available

CITED BY

None

KEYWORDS

IEEE Keywords

Cloud computing, Context, Guidelines, Measurement, Monitoring, Security, Standards

CORRECTIONS

None

Personal Sign In | Create Account

IEEE Account

» Change Username/Password

» Update Address

Purchase Details

» Payment Options

» Order History

» View Purchased Documents

Profile Information

» Communications Preferences

» Profession and Education

» Technical Interests

Need Help?

» US & Canada: +1 800 678 4333

» Worldwide: +1 732 981 0060

» Contact & Support

About IEEE *Xplore* | Contact Us | Help | Terms of Use | Nondiscrimination Policy | Sitemap | Privacy & Opting Out of Cookies