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

Valentina Casola*, Alessandra De Benedictis*, Mădălina Eraşcu†, Massimiliano Rak‡ and Umberto Villano§
*DIETI, University of Naples Federico II, Naples, Italy
Email: {alessandra.debenedictis, casolav}@unina.it
†Institute e-Austria, Timişoara, Romania
Email: merascu@info.uvt.ro
‡DII, Second University of Naples, Aversa, Italy
Email: massimiliano.rak@unina2.it
§DING, University of Sannio, Benevento, Italy
Email: villano@unisannio.it

***Abstract*—The extensive use of cloud services by both individual users and organizations induces several security risks. The risk perception is considerably increased when a Cloud Service Provider (CSP) fails in clearly stating its security policies, and when a definite mapping between user-defined security requirements and the *security capabilities* offered by the provider is missing. Service Level Agreements that include security-related guarantee terms (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 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.**

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 as well as individual customers rely upon cloud services to carry out their business in an efficient and effective way. 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. The less transparent a Cloud Service Provider's (CSP's) security policies are, the higher the risk is perceived. In spite of the very low costs, the lack of security and even the lack of security perception may have a negative influence on CSPs' business goals, in addition to representing at present the stronger limitation in the adoption of cloud services for those customers that need security guarantees, for example, due to regulatory compliance issues.

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. This represents a limitation both for customers, who lack the tools to wisely choose the services to acquire based on their security properties, and for CSPs, who miss the opportunity to correctly locate their offers with respect not only to different customers' requirements, but also to security-related regulations. Indeed, security technologies and mechanisms are mature enough to be offered as all the other functionalities, namely *as-a-Service*, and time is mature to think of security as something that can be negotiated, evaluated, acquired and covered by a Service Level Agreement as any other service. Nevertheless, the adoption of SLAs in the Cloud environment and, above all, the adoption of Security SLAs seems to be still a holy grail.

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]. The most interesting issues related to the adoption of Security SLAs are the identification, quantification (in terms of security level) and monitoring of security parameters associated with existing offers that can be easily understood and monitored by customers [2]. Related to identification and quantification, several guidelines and international standardization initiatives exist, which aim at defining a shared catalogue of *security controls* related to both technical and non technical aspects [3], [4], [5]. Such security control frameworks are mainly intended to be used by organizations to assess the level of security of their services by specifying the security controls

that the provided *security capabilities*[1] are able to enforce, and some approaches exist in the literature that try to match the customers' requirements and the providers' offers in terms of security capabilities. Some of them use techniques such as ontologies to map security controls with terms of Security SLAs [6], [7], [8], to enable automatic negotiation of security and comparison among different offers. Others focus on the quantitative evaluation of security based on the definition of security metrics [9] or analytical approaches [10].

However, despite the strong interest in security and the existing efforts towards standardization, Security SLAs are far from being commonly adopted by current CSPs. Indeed, most CSPs only report performance-related parameters in their SLAs, and 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. In the last years, several projects have been devoted to security-driven design of Cloud applications (e.g., MODAClouds, Contrail, PoSecCo, A4CLOUD, MUSA).

The authors of this paper are involved in two EU projects (SPECS[2] and MUSA[3]) 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. The SPECS framework comes into play to enhance the offers of existing providers by means of the activation of security capabilities that can be negotiated by customers, automatically enforced through an enriched supply chain[4], and continuously monitored according to a signed Security SLA. In this paper, we focus on the construction of the supply chains involved in the delivery of security-enhanced services. We propose, and illustrate through a case study, 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 two different security capabilities offered through the SPECS framework and, finally, Section VI draws our conclusions.

[1] A security capability is defined by the NIST as *a combination of mutually-reinforcing security controls (i.e., safeguards and countermeasures) implemented by technical means (i.e., functionality in hardware, software, and firmware), physical means (i.e., physical devices and protective measures), and procedural means (i.e., procedures performed by individuals)* [4].

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

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

[4] A supply chain is represented by a set of available components able to cover the security requirements expressed by a customer, i.e. the security capabilities he needs, along with their configurations.

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 [11] 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.

Extensive work has been recently done on Security SLAs, especially related to cloud environments, in the context of both academical research and industry and government-driven initiatives. In 2011, ENISA published a report analyzing the use of security parameters in cloud SLAs (mostly focused on the EC public sector) [12]. This report was based on a survey of real-world CSP SLAs, and listed a set of security SLOs commonly found there. It 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, while merely security-related parameters were not included. 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 [13] 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 2014, the C-SIG SLA subgroup, an industry group facilitated by the European Commission, has released a set of SLA standardization guidelines [14] for CSPs and professional cloud customers, which provide definitions of the legal and technical terms used in SLAs and identify specific SLOs designed to achieve standardization for several aspects of SLAs, including secure data management and protection. The C-SIG guidelines, which have been submitted to the ISO Cloud Computing Working Group to be taken into account by the ISO/IEC 19086 international standard [15], aim at specifying measurable security level objectives in order to enable the management of security from the perspective of both what is offered by providers and what is requested by customers. The identification of the security level objectives applicable to a service is just related to the specification of the security controls that the provider is able to implement related to such service, namely to the security capabilities offered on top of it. This is the biggest limitation of these approaches, as they do not take into account the different constraints associated to the need of guaranteeing the security provided by these capabilities. 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 [16], released in 2009 and based on ISO 27001/2 standards 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. It included a set of questions to submit to providers, related to 10 different security aspects. Security domains have been improved and enriched by several subsequent standards, such as the current versions of ISO 27001/2, namely ISO 27001/2:2013 [17], [3], which define 114 controls in 14 groups and provide a specification for Information Security Management Systems (ISMSs) that may be used to gain an official certification issued by an independent and accredited certification body on successful completion of a formal audit process.

Similarly, NIST sp800-53 [4] structures the controls across a three level-hierarchy: in order to simplify the security control selection and specification process, controls are organized into 18 families or categories, each containing security controls related to the general security topic of the family, such as access control, audit and accountability, incident response etc. Moreover, some controls may have a set of associated control enhancements, containing supplemental guidance. NIST's security controls and control enhancements have been developed and integrated that address areas such as mobile and cloud computing, applications security, trustworthiness, assurance, and resiliency of information systems, insider threat, supply chain security and the advanced persistent threat. Finally, the Cloud Control Matrix [5] 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. In this scenario, many security issues arise since, as simply illustrated in Figure 1, security vulnerabilities may be exploited at user, communication and service (IaaS, PaaS and SaaS) level. With respect to the above discussed issues, our work (in the context of the SPECS project) 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 (cf. Figure 2). It especially addresses the needs of medium/small CSPs willing to broaden their business opportunities by trying to meet security requirements of prospective customers, which however do not have the capability to take on all the respective burden. 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.

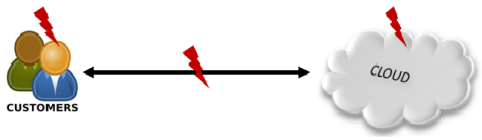

Fig. 1. Security issues in the cloud

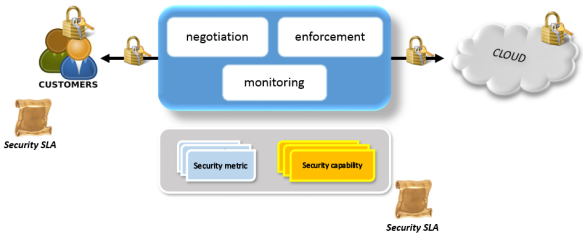

Fig. 2. Automatic enforcement of SLA-based Security Services

III. Enforcing Security-as-a-Service: the SPECS approach

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 [18], [19]. In particular, the SPECS framework exploits state-of-the-art PaaS solutions such as cloud4SOA [20], Contrail [21], mOSAIC [22], Google App Engine and Microsoft Azure, and enhances them by building a PaaS offering security (Security-as-a-Service) through an SLA-based approach.

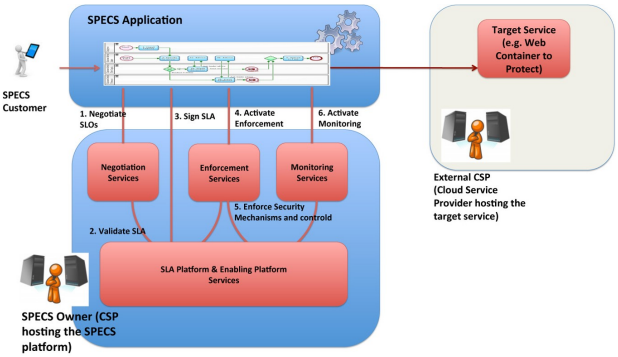

Fig. 3. The SPECS approach

As illustrated in Figure 3, 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, each representing a different supply chain to implement, 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)*. Indeed, given a set of security requirements expressed by the SPECS Customer, more than one supply chain may be identified, each characterized by its own cost and associated level of security, and the resulting supply chains may be ranked to help the SPECS Customer choose the desired configuration. 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 [23], and is a fundamental step toward fostering the actual adoption of Security SLAs for cloud service.

IV. Enabling the Automatic Enforcement of Security

As discussed in Section II, several approaches exist in the literature which 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 4):

1) identification of the security capabilities that can be offered on top of the considered cloud services;
2) analysis of reference *security control frameworks* to identify the *control categories* and the baseline *security controls* that can be applied to the considered cloud services by implementing the security features defined at the previous step. The identified controls are collected in *security capabilities* and implemented by proper components;
3) identification of monitorable metrics and/or enforceable parameters associated to each control defined at the previous step. 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) identification of the set of admissible values for each monitorable metric/enforceable parameter and definition of possible SLOs on top of them.

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.

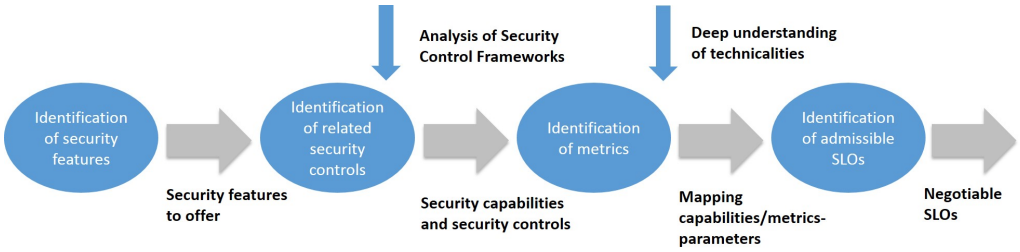

Fig. 4. Methodology overview

In Section V, we will illustrate an example of application of this methodology for a case study represented by a provider offering different security capabilities and security SLAs to provision a secure web container service.

V. A Case Study

In order to illustrate the introduced approach, in this section we discuss a case study involving the provisioning of a web container service. The web container is represented by one or more Virtual Machines (VMs) deployed on the CSP's IaaS resources, for which specific security guarantees related to the protection of the communication channel and to the resiliency to attacks and failures are requested. We assume the web container service is offered with no security guarantees by a CSP, while desired security capabilities, as well as related monitoring services, are provided by the SPECS framework and can be dynamically enforced in the web container service's supply chain based on customers' requirements. According to the proposed methodology, in Section V-A, we illustrate the process followed to set-up a set of negotiable capabilities 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.

In this section, we discuss the considered capabilities and identify the related security controls and the associated security metrics/enforceable parameters, as devised in the steps of the proposed methodology. In this case study, we consider the following two security features offered by the SPECS framework[5] (***Step 1 of methodology***):

- *TLS/SSL* [24], enforcing a secure channel on top of communications involving the web container;
- *Web Container Pool*, ensuring the resiliency to attacks and failures through the acquisition of a set of VMs (e.g., by means of a component which acts as a broker) which are configured with different web container instances in order to guarantee a negotiable level of diversity and redundancy.

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 Tables I and II, which list control families, associated security controls and control enhancements related to TLS/SSL and Web Container Pool security features respectively. These sets of controls represent the *TLS/SSL*and the *Web Container Pool* capability respectively.

A non-exhaustive list of monitorable metrics and enforceable parameters for the TLS/SSL and Web Container Pool capabilities, as a result of ***Step 3 of methodology***, is reported in the following:

TLS/SSL Security Capability.
For the *TLS/SSL* capability, we identified the following list of metrics:

- *Cryptographic Strength* [25] is a measure of the expected number of operations required to defeat a cryptographic mechanism.
- *Forward Secrecy*[6] is a property ensuring that a session key derived from a set of long-term keys cannot be compromised if one of the long-term keys is compromised in the future.
- *HSTS (HTTP Strict Transport Security)* [26] defines a mechanism enabling web sites to declare themselves accessible only via secure connections and/or for users to be able to direct their user agent(s) to interact with given sites only over secure connections.
- *HTTP to HTTPS Redirects* is the most common requirement on most servers which ensures that the connections are coming from customers using SSL.
- *Secure Cookies Forced* is a measure enabling the use of secure cookies.
- *Client Certificates* are used to digitally identify a particular individual or user with an authentication server, in our case, TLS/SSL-based authentication.
- *Certificate Status Request (OCSP stapling)* [27] allows the presenter of a certificate to bear the resource cost involved in providing OCSP responses, instead of the issuing certificate authority.

[5]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 these two for brevity reasons.

[6]https://community.qualys.com/blogs/securitylabs/2013/06/25/ssl-labs-deploying-forward-secrecy

TABLE I. SECURITY CONTROLS APPLICABLE TO TLS/SSL

Control Framework	Control Family	Security Control	Control Enhancement
NIST-800-53r4	SC - System and Communication Protection	SC-8 Transmission Confidentiality and Integrity	SC-8(1) CRYPTOGRAPHIC OR ALTERNATE PHYSICAL PROTECTION SC-8(2) PRE / POST TRANSMISSION HANDLING SC-8(3) CRYPTOGRAPHIC PROTECTION FOR MESSAGE EXTERNALS
		SC-12 Cryptographic Key Establishment and Management	SC-12(1) AVAILABILITY SC-12(2) SYMMETRIC KEYS SC-12(4) PKI CERTIFICATES SC-12(5) PKI CERTIFICATES / HARDWARE TOKENS
		SC-13 Cryptographic Protection	SC-13(1) FIPS-VALIDATED CRYPTOGRAPHY SC-12(2) NSA-APPROVED CRYPTOGRAPHY SC-13(3) INDIVIDUALS WITHOUT FORMAL ACCESS APPROVALS SC-13(4) DIGITAL SIGNATURES
		SC-17 Public Key Infrastructure Certificates	N/A
		SC-23 Session Authenticity	SC-23(1) INVALIDATE SESSION IDENTIFIERS AT LOGOUT SC-23(3) UNIQUE SESSION IDENTIFIERS WITH RANDOMIZATION SC-23(5) ALLOWED CERTIFICATE AUTHORITIES
		SC-43 Usage Restrictions	N/A

- *Certificate Pinning* [28] defines a new HTTP header that enables user agents to determine which Subject Public Key Info structures will be present in a web host's certificate chain in future TLS/SSL connections.
- *DANE*[7] is a metric which, in case of secure connections via TLS/SSL, ensures that one is using the correct TLS/SSL certificate.
- *FIPS Compliance*[8] is a metric which, if adopted, ensures that TLS/SSL is compliant with FIPS. Federal Information Processing Standards (FIPS) are United States Government standards that provide a benchmark for implementing cryptographic software, specifying best practices for implementing crypto algorithms, handling key material and data buffers, and working with the operating system.

As anticipated, some of the listed items are actually metrics which are monitorable in case of HTTP connections (e.g., Cryptographic Strength, Forward Secrecy, Client Certificates, Certificate Status Request, FIPS Compliance), others are configuration parameters that increase the level of security of an already adopted HTTPS connection (e.g., HSTS, HTTP to HTTPS Redirects, Secure Cookies Forced, Certificate Pinning), while others need preliminary configurations which would permit their monitoring (e.g., DANE).

Web Container Pool Security Capability.
Regarding the Web Container Pool capability, we considered the two following related 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.

We also filled out a list of possible values associated to each of these metrics/parameters for both capabilities (cf. Tables III and IV) that, as discussed previously, are exposed to customers during negotiation for the selection of SLOs and are used during enforcement to configure the security capabilities accordingly (***Step 4 of methodology***). To the best of our knowledge, such a comprehensive list has never been taken in consideration for negotiation and enforcement purposes. We came up with it by analyzing the latest recommendations of specialized organizations [9] or by a deep understanding of the definitions of the metrics. Finally, Table V and Table VI summarize the mapping between security controls identified for each capability (cf. Tables I and II) and respective metrics.

Note that the described mapping is very challenging and requires deep understanding of the security standards and specific technicalities related to select security capabilities, hence we do not claim that the application our methodology is complete, while it can be improved for sure. 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, the reported mapping 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

[7]http://www.internetsociety.org/deploy360/resources/dane/
[8]http://www.nist.gov/itl/fips.cfm

[9]For Cryptographic Strength we proposed to use ECRYPT II level [29], which rates the strength of an algorithm on a scale from 1 to 8, based on the key length, algorithm and/or output length.

TABLE II. SECURITY CONTROLS APPLICABLE TO WEB CONTAINER POOL

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 III. TLS/SSL METRICS/PARAMETERS AND POTENTIAL VALUES

Metric Name	Potential Metric Value
Cryptographic Strength	Level 1 $<$ ... $<$ Level 8
Forward Secrecy	Required / Preferred / Forbidden
HSTS	Yes / No
HTTP to HTTPS Redirects	Yes / No
Secure Cookies Forced	Yes / No
Client Certificates	Required / Preferred / Forbidden
Certificate Status Request	Yes / No
Certificate Pinning	Yes / No
DANE	Yes / No
FIPS Compliance	Yes / No

TABLE IV. WEB CONTAINER POOL METRICS/PARAMETERS AND POTENTIAL VALUES

Metric Name	Potential Metric Value
Level of Redundancy	1, 2, 3, ...
Level of Diversity	1, 2, 3, ...

TABLE V. MAPPING BETWEEN SECURITY CONTROLS AND TLS/SSL METRICS

Security Control	SSL/TLS Metrics
SC-8	HTTP to HTTPS redirects
	Client Certificates
SC-12	Forward Secrecy
SC-13	Cryptographic Strength
	FIPS Compliance
SC-17	Certificate Status Request
	Certificate Pinning
	DANE
SC-29	Secure Cookies Forced
SC-43	HSTS

TABLE VI. MAPPING BETWEEN SECURITY CONTROLS AND WEB CONTAINER POOL METRICS

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

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.

B. Negotiating a secure web container service

A SPECS Customer represented by a web developer aims at acquiring a reliable and secure web container from an IaaS provider (called the External CSP from now on). 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 (TLS/SSL, authentication and authorization protocols and so on). 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 capabilities, namely TLS/SSL and Web Container Pool, whose set-up has been illustrated previously. The SPECS Customer selects both capabilities, 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 agreed metrics/parameters (e.g., `Cryptographic Strength=3` and `HTTP to HTTPS redirects=yes` for TLS/SSL and `level of diversity=2` and `level of redundancy=3` for Web Container Pool). SPECS acquires the needed IaaS resources from the CSP on behalf of the SPECS Customer (registered on SPECS), and sets-up and activates the components devoted to implement the Web Container Pool and TLS/SSL capabilities.

Scenario 2. A SPECS Customer expert in security accesses the wizard offered by the SPECS Application and is prompted with a description of the available capabilities, namely TLS/SSL and Web Container Pool. The SPECS Customer selects both capabilities, and is prompted with the list of associated security controls, as illustrated in Tables I and II. The SPECS Customer is interested in the *System and Communications Protection (SC)* category, and i particular in the SC-8 (Transmission Confidentiality and Integrity) control belonging to the TLS/SSL capability and the SC-29 (Heterogeneity) control associated with the Web Container Pool capability. The wizard returns the metrics/parameters associated to these two controls (cf. Tables V and VI) with respective admissible values (cf. Tables III and IV). The SPECS Customer selects the following: `Cryptographic Strength=3` and `HTTP to HTTPS redirects=yes` for TLS/SSL and `level of diversity=2` and `level of redundancy=3` for Web Container Pool. 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.

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 EC FP7 SPECS Project, and 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 two different security capabilities 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).

References

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

[2] V. Casola, A. De Benedictis, and M. Rak, "Security monitoring in the cloud: An SLA-based approach," in *Availability, Reliability and Security (ARES), 2015 10th International Conference on*, Aug 2015, pp. 749–755.

[3] International Organization for Standardization, "ISO/IEC 27002:2013 Information Technology, Security Techniques, Code of Practice for Information Security Management," 2013.

[4] National Institute of Standards and Technology, "NIST SP-800-53: Recommended Security Controls for Federal Information Systems," 2013.

[5] Cloud Security Alliance, "Cloud Control Matrix v3.0," https://cloudsecurityalliance.org/download/cloud-controls-matrix-v3/.

[6] L. Liccardo, M. Rak, G. Di Modica, and O. Tomarchio, "Ontology-based negotiation of security requirements in cloud," in *Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference on*, Nov 2012, pp. 192–197.

[7] M. Hale and R. Gamble, "Building a compliance vocabulary to embed security controls in cloud SLAs," in *Services (SERVICES), 2013 IEEE Ninth World Congress on*, June 2013, pp. 118–125.

[8] C.-Y. Lee, K. M. Kavi, R. A. Paul, and M. Gomathisankaran, "Ontology of secure service level agreement," in *High Assurance Systems Engineering (HASE), 2015 IEEE 16th International Symposium on*, Jan 2015, pp. 166–172.

[9] V. Casola, A. Mazzeo, N. Mazzocca, and M. Rak, "A SLA evaluation methodology in service oriented architectures," *Advances in Information Security*, vol. 23, pp. 119–130, 2006, cited By 26. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84882690392&partnerID=40&md5=45028debde72bba468bce54ec2a16829

[10] V. Casola, A. Fasolino, N. Mazzocca, and P. Tramontana, "An AHP-based framework for quality and security evaluation," vol. 3, 2009, pp. 405–411.

[11] R. R. Henning, "Security service level agreements: Quantifiable security for the enterprise?" in *Proceedings of the 1999 Workshop on New Security Paradigms*, ser. NSPW '99. New York, NY, USA: ACM, 2000, pp. 54–60. [Online]. Available: http://doi.acm.org/10.1145/335169.335194

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

[13] ENISA, "Procure Secure. A guide to monitoring of security service levels in cloud contracts," April 2012.

[14] European Commission – C-SIG (Cloud Select Industry Group) subgroup, "Cloud Service Level Agreement Standardisation Guidelines," Tech. Rep., June 26 2014.

[15] "ISO/IEC NP 19086-1. Information Technology–Cloud computing–Service level agreement (SLA) framework and technology–Part 1: Overview and concepts," International Organization for Standardization, Tech. Rep., 2014.

[16] ENISA, "Information Assurance Framework," November 2009.

[17] International Organization for Standardization, "ISO/IEC 27001:2013 Information technology Security techniques Information security management systems Requirements," 2013.

[18] V. Casola, A. De Benedictis, M. Rak, and U. Villano, "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, Barcelona, Spain, April 3-5, 2014.*, 2014, pp. 752–757.

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

[20] D. Zeginis, F. D'Andria, S. Bocconi, J. G. Cruz, O. C. Martin, P. Gouvas, G. Ledakis, and K. A. Tarabanis, "A user-centric multi-PaaS application management solution for hybrid multi-cloud scenarios," *Scalable Computing: Practice and Experience*, vol. 14, no. 1, 2013.

[21] G. Pierre and C. Stratan, "ConPaaS: A platform for hosting elastic cloud applications," *IEEE Internet Computing*, vol. 16, no. 5, pp. 88–92, 2012.

[22] D. Petcu, B. D. Martino, S. Venticinque, M. Rak, T. Mahr, G. E. Lopez, F. Brito, R. Cossu, M. Stopar, S. Sperka, and V. Stankovski, "Experiences in building a mosaic of clouds," *Journal of Cloud Computing*, vol. 2, no. 1, 2013.

[23] V. Casola, A. De Benedictis, M. Rak, J. Modic, and M. Erascu, "Automatically enforcing security slas in the cloud," *IEEE Transactions on Services Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[24] National Institute of Standards and Technology, "NIST SP-800-52: Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations," 2014.

[25] ——, "NIST SP-800-63: Information Security," 2006.

[26] J. Hodges, C. Jackson, and A. Barth, "HTTP Strict Transport Security (HSTS)," Carnegie Mellon University & Google, Inc., Tech. Rep., 2012.

[27] P. Hallam-Baker, "X.509v3 Extension: OCSP Stapling Required draft-hallambaker-muststaple-00," Comodo Group Inc., Tech. Rep., 2012.

[28] C. Evans, C. Palmer, and R. Sleevi, "Public Key Pinning Extension for HTTP draft-ietf-websec-key-pinning-21," Google, Inc., Tech. Rep., 2014.

[29] European Network of Excellence in Cryptology II - National Institute of Standards and Technology, "ECRYPT II Yearly Report on Algorithms and Keysizes (2011-2012)," 2012.