A FRAMEWORK OF SECURITY ORCHESTRATION TO MANAGE CLOUD COMPUTING SLA

Shikha Singh[1], Dr. Ajay Kumar Bharti[2], Dr. Himanshu Pandey[3], Dr. Durgansh Sharma[4], Dr. N.R.Shanker[5]

IEEE student ,Research Scholar, Computer Science and Engineering[1], Professor,School of Computer Application[2], Assistant Professor, Computer Science and Engineering[3], Associate Professor,School of Business Management[4], Professor,Computer Science and Engineering[5]

Maharishi University of Information and Technology, Lucknow,U.P.,India[1],Babu Banarasi Das University, Lucknow,U.P.,India[2], Lucknow University, Lucknow,U.P.,India[3],,Christ University Ghaziabad,U.P.,India[4], Aalim Muhamed Salegh College of Engineering, Chennai,Tamil Naidu,India[5]

ABSTRACT

The cloud offers intriguing possibilities for relocating company applications, and the corporate security manager may rest easy knowing that no on-premises hardware requires monitoring or protection. Providers of cloud services and the ecosystems that support them are constantly innovating to keep up with the growing demand for cloud computing. This includes new types of services, new ways of delivering those services, and new ways of working together. SLA management's benefits include being able to negotiate appropriate service parameters, putting in place mechanisms to assure that service execution in the external cloud is in line with agreed SLAs, and monitoring to verify compliance by the cloud provider. Despite rapid advancements in both theory and practice, the legal/contractual, economic, service quality, interoperability, security, and privacy challenges still provide considerable obstacles to the widespread use of cloud computing. Several models for deploying cloud services are described. In order to make cloud storage more effective, we present a SLA-aware resource method in this work. Our approach maximizes both storage space usage and I/O throughput at the backend nodes. Customers are offered the present Cloud setup on a best-effort basis. Rather than making any guarantees, a statistical uptime expectation is shared with the user, and minor compensations are offered in the event of any unscheduled downtime.

Key words: Cloud Computing, SLA, Security, Orchestration, Algorithm

INTRODUCTION

The phrase "cloud computing" is used to describe any method of providing hosted services through the internet. Infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) are the three primary classifications of cloud computing services (SaaS). One's cloud of choice might be either public or private. Anyone with access to the internet may purchase services from a public cloud. Private clouds are secure, isolated networks or data centers that only allow authorized users to access the resources they need.

Cloud computing, whether it's a public or private system, aims to make it simple and affordable to access a wide range of IT resources and services on demand.

Cloud infrastructure refers to the collection of tools, including computer programs and servers, that are necessary to put into action a cloud computing architecture. Utility computing and on-demand computing are two more names for cloud computing. Service Level Agreements (SLAs) are crucial to the service lifecycle because they capture service expectations and entities' duties, which in turn informs engineering choices at the conception level (during, for example, service design) and operational decisions at the delivery level (during for example service usage and delivery). Service level agreements (SLAs) allow the parties involved to come to terms on the services to be provided, the means by which they will be provided, and the responsibilities of each party in the event of any problems.

Despite the many benefits of cloud computing, this computing paradigm still faces a number of challenges that must be addressed before it can be widely used. First, the user may not have the power to audit or alter the procedures and rules under which he or she must operate in cloud computing, or even the level of control over data or application performance that is necessary. If an application's components are spread out across various regions of the cloud, it might slow down. Cross-border compliance rules are particularly complex, and it's worth noting that more comprehensive cloud computing legislation are still in the works. It seems to reason that managing remote computers is more complicated than managing local Intranet PCs.

Still, Service Level Agreements are just contracts that spell out what will be expected of whom. A service level agreement (SLA) cannot ensure delivery of the promised service any more than a car warranty can ensure your vehicle will never experience mechanical failure, as stated in. In instance, a service level agreement (SLA) cannot improve upon a subpar offering. However, a service level agreement might reduce the danger of picking a poor provider. The latter emphasizes the significance of supplementary tools and procedures utilized at various stages of the SLA lifecycle, such as monitoring service execution conformance to the agreed conditions and enforcement through triggering of actions to meet evolving needs. The primary focus of these kinds of frameworks is to guarantee that the service is of a certain quality (as set by the corresponding QoS attributes). Several players in the cloud computing industry have voiced the following urgent need: While one cloud provider, Cloud One, notes that "much good work has been completed on SLAs and the entire business model around the cloud, but much remains," other researchers have noted the importance of SLAs when sending sensitive data offsite, including analysts from Forrester and Accenture. Typically, it is up to the customer to identify service level agreement (SLA) violations, contact the service provider, and obtain a credit.

Cloud service level agreements (SLAs) may help CSPs in more ways than one. A CSP is compelled to deal with security issues because of the precise nature of the security criteria specified in SLAs. End customers may also get knowledge of the costs and advantages of this new service model thanks to SLAs. SLAs are designed to supplement existing assurance methods for security policy enforcement, but it is also apparent that they are not meant to replace them. ENISA has also acknowledged the significance of Cloud SLAs, highlighting the need for the European Union to produce template contracts and service level agreements (SLAs). According to a recent poll by ENISA, a large percentage of Cloud clients do not perform ongoing monitoring of the security components of their agreed SLA. Because of this, patrons are in the dark regarding several crucial features of service security. A security breach may be the point at which they learn about the ineffectiveness of their security procedures. According to the results of the study, security criteria are less adequately covered by SLAs than availability concerns, despite the fact that SLAs are widely employed. According to Bernsmed, a Cloud SLA mimics the CSP's security at the service level and is based on either a set of expert-driven security standards (for example, for compliance reasons) or some form of preparatory threat analysis. As suggested in several commercial and academic books, e.g., this process yields a set of security statements (also known as security provisions) in the form security attribute, value (such as Backup Frequency, Daily and Encryption Key Size, 512bits). These security measures should be broken down into manageable categories using a taxonomy like Savola or the Cloud Controls Matrix developed by the Cloud Security Alliances. Finally, the Cloud SLA template will be complete, and CSPs will be able to utilize it to define their own SLAs based on this collection of security requirements that has been categorized into taxonomic categories. Cloud service level agreements (SLAs) are often kept in reputable and easily accessible public databases like the Cloud Security Alliances' (CSA) Security, Trust & Assurance Registry (STAR). Although Almorsy and Luna highlight the difficulties in establishing SLAs for actual Cloud deployments, the existing dearth of methods to statistically reason about them has also been shown to be an impediment to SLA use.

Hundreds of cloud users may access thousands of IoT services in a multi-cloud environment supported by the Internet of Things. When it comes to authenticating users in a multi-cloud environment, the standard authentication methods used in conventional networks fall short. Because of this, this article presents a lightweight and unique solution for the dynamic authentication that allows users to sign in once while connecting to a foreign cloud. While still being secure, the suggested authentication system outperforms common methods like SAML and Kerberos. Next, we provide an algorithm for selecting the most suitable Internet of Things service from among a number of cloud vendors in terms of meeting the quality-of-service criteria of individual users (QoS). Next, service level agreements (SLAs) are implemented to manage service execution in the foreign cloud and guarantee security. Users' functional (CPU, RAM, memory, etc.) and non-functional needs (bandwidth, latency, availability, dependability, etc.) for a specific IoT service may be negotiated via the use of SLA mechanisms, allowing for safe cooperation amongst multi-clouds. The onus of enforcing methods that meet the QoS standards agreed upon in the SLA will be on the multi-cloud that is receiving user requests. In contrast, the SLA's monitoring phase entails keeping tabs on how an IoT service is being deployed in a different country's cloud to ensure it's meeting the terms of the agreement.

LITERATURE REVIEW

Halabi, Talal et.al. (2018) Concerns about data security continue to deter firms that handle sensitive data from making the move to the cloud. In recent times, Security Service Level Agreements have been used in an attempt to define the security features of a Cloud service (Security-SLAs). However, it might be difficult to measure and keep tabs on Security-SLAs in their current form and under their existing parameters. A faster rate of Cloud adoption and an increase in consumers who can feel safe using Cloud computing's benefits thanks to standardized, quantifiable Security-SLAs are assured. This article proposes a broker-based system for controlling the Cloud Security-SLA. In order to better portray the agreement, we first create a standardized, quantifiable, and measurable form. Then, we present an assessment and selection model where the primary focus is on finding the best possible balance among competing objectives, such as the three pillars of security (CI, A), using a multi-objective optimization problem. The simulation results show the Pareto optimum set of options and how the consumer may choose the best service provider based on the nature of the service and the cost of the service.

Silva, Carlos Alberto (2015) The fact that Cloud Computing is the meeting place of numerous emerging technologies may explain why no new forms of security have yet to be developed for the cloud. These technologies include Utility Computers, Computational Grids, Autonomous Computing, Virtualization, and Service Oriented Architectures. Each current cloud security control was mapped to several controls from the preexisting, general-purpose control frameworks, although these fundamental regions have been separately handled by existing general-purpose security controls. We also found that procedures for specification, monitoring, and security management in cloud settings are in high demand. We hypothesize that a Security Service Level Agreement (Security-SLA) may be useful in this situation for specifying security restrictions that are tailored to the requirements of service end-users. The Security-SLA might then be automated to increase security.

OrazioTomarchio (2020) As the number of businesses offering cloud services and the number of people using them both continue to grow, it's clear that this model is living up to its potential. Despite this, cloud players are feeling the burden of the industry's fast growth. Maximizing profits while maintaining high levels of customer satisfaction is a major concern from the provider's perspective. Customers have a challenging choice between competing offers of similar services from different vendors. Several studies have shown that software frameworks (termed CROFs, for Cloud Resource Orchestration Framework) may be used to orchestrate the disparate resources of several cloud providers in a way that best meets the demands of the client. In addition to highlighting the multi-cloud computing open challenges that need to be addressed in the near future, we believe this work will serve as a complete analysis and comparison of the most relevant CROFs discovered in the literature.

PetarKochovski (2022) New intelligent apps that rely on data from IoT devices generally need cloud storage to work properly. The high Quality of Service (QoS) needs of these applications need a hybrid Edge/Fog computing architecture. Few users of storage services ever think to ask for guarantees like SLAs (SLAs). We address these challenges by providing containerized storage services in the form of Fog Storage Services that include QoS. Quality of service measures, such as high availability and fast throughput, are used in conjunction with a novel Pareto-based decision procedure to dynamically identify where data storage containers should be located along the Things-to-Cloud continuum. The proposed strategy is first tried out in a virtual environment. It has been shown that the DECENTER Fog and Brokerage Platform can be used to orchestrate storage containers with the help of the decision-making process and a new service level agreement specification. The proposed decision-making technique has been validated by simulation and real-world testing in a pan-European testbed, demonstrating that it provides a set of optimal storage nodes that meet the SLA requirements. New intelligent applications may be provided for our planned Fog Storage Services with the SLA assurances and acceptable QoS we need.

Ficco, M. et.al. (2012) Computing in the cloud is a novel approach. Service Level Agreement management and security management are two of the many difficulties in this area that are regarded as particularly important. In business parlance, an SLA is a contract between a service provider and their clientele outlining certain service-related terms and conditions. Its purpose is to provide an uncomplicated framework within which end users and the Service Provider may reach an agreement outlining the precise quality guarantees that will be implemented. With cloud computing, resources like servers and software are hosted remotely and made available to users on demand through a web interface and a pay-as-you-go pricing model. This article aims to demonstrate how a Cloud-oriented API generated from the mOSAIC project can be used to construct a SLA-oriented Cloud application that allows the delivery of security solutions as a service. Our attention will be drawn to intrusion tolerance solutions, or those that allow a system to continue operating with (at least some) degree of availability even if a security breach has occurred.

RESEARCH METHODOLOGY

Service Level Agreements

The WS-Agreement for Java implementation (WSAG4J) developed by the SCAI at the Fraunhofer Institute is the basis of the OPTIMIS Cloud SLM module. WSAG4J's parts are shown in Figure 1, and their short descriptions may be found below.

– The API Module

Interface definitions and implementations common to several framework modules are collected here (not shown in the Figure 1 since it is used by all modules). – WSAG4J's Client-Side Java API the API module defines the client API, which is implemented here.

Used to connect to the WSAG4J web service stack.

– The SLA Engine Module

The WSAG4J framework revolves on this core module. It's an example of a generic SLA engine based on Web Services Agreement (WS-Agreement). It implements the typical features for handling offers, generating contracts, tracking contracts in action, and assessing and accounting for guarantees. The business logic required to instantiate and track SLA-aware services, as well as any regulations governing acceptance of such agreements, may be plugged in with little effort.

– TheWSAG4J Web Service Module

The WSAG4J engine's remote user interface is realized inside this module. The WS-Agreement port types are implemented, and WSAG4J is called upon to do any required processing. The Apache Muse framework [8] serves as the backbone for the web service module. With this framework, you may build the WSRF container called for by the WS-Agreement standard.

– The Server Distribution

The WSAG4J server, comprising the web service stack, the SLA engine, and all necessary settings, are part of this component. It's a web app archive that can be deployed quickly and simply across several servers.

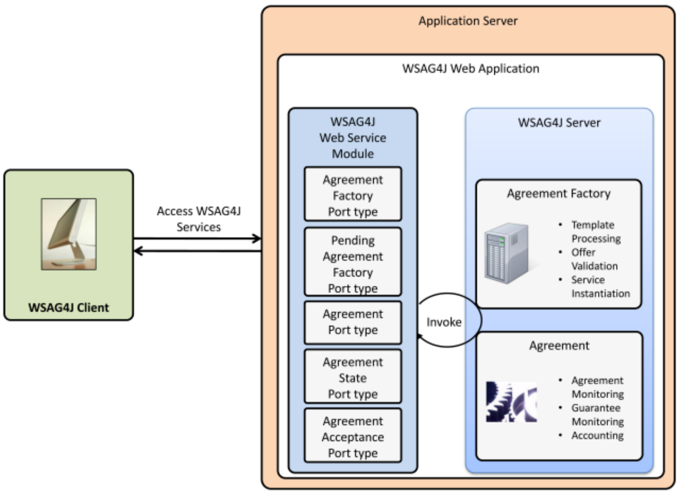

Figure1. Components of WSAG4J

Cloud Computing Security Management

Implementing Cloud-Based Security Engineering Security requirements for web services and cloud web applications built from web services were presented by Menzel et al., who also provided a model-driven strategy and language for doing so. Each instance of an app (and its associated services) runs on its own virtual machine. They made the assumptions that (1) online applications are made up entirely of web services, (2) multi-tenant security is maintained by the use of individual virtual machines for each tenant (the simplest scenario), and (3) the security of the underlying infrastructure is ignored. With a similar but more abstract approach, Bertram et al. advocated security engineering for cloud hosted services (risk-based instead of security-requirements based). The authors made the safe assumption that two cooperating businesses would be using a secure cloud platform that would include security PaaS to monitor and reduce threats to the shared services. Web services are the exclusive focus of both projects, and security is captured and generated at the service level rather than the lower levels.

Cloud Security Management

A quantitative risk analysis and assessment approach, based on NIST- FIPS-199, was suggested by Saripalli et al. Part of the SMP process involves analyzing potential dangers. Still needed are the rest of the SMP's procedures. Despite proposing a quantitative technique, the authors only used qualitative rating bands while conducting their own risk assessments (Low, Medium, and High). Xuan et al. also made comparable attempts. The two most common ISMS standards are ISO27000 and NISTFISMA. Since both imply the asset owner is in complete command of the SMP, they are incompatible with a cloud infrastructure (hosted inside enterprise boundaries). Plus, they don't take into account the "Multitenancy" situation, when several users use the same service. Associated studies in ISMS investigate risk assessment and management frameworks like OCTAVE, CORAS, and others. There are a number of different types of security management systems, including policy-based security management, a hybrid of ontology-based and policy-based management, and model-based security management. Most of these methods overlook the SMP's feedback and improvement stages in favor of the security capture and enforcement stages. This is especially true in the cloud model, where we must now worry not just about protecting data inside our own networks but also about the safety of data stored on other servers.

Cloud Security SLA management

Another method for defining and administering security is the use of a security service level agreement (SLA). Despite the many suggestions made in SLA management (SLA design, enforcement, and monitoring), security is seldom taken into account since it is unique among the other QOS qualities. The goals of Shirlei et alSec- SLA.'s were limited to those having to do with the backup strategy for data. Despite the cloud SLA management architecture presented by Pankesh et al., security is not addressed. The difficulty in establishing adequate security measurements is a valid rationale for the absence of SecSLAs.

SLA-Aware Scheduling Algorithm

Our effort has two main objectives: first, to optimize scheduling in order to get the most out of each storage node in terms of I/O throughput SLA usage, and second, to have I/O throughput management implemented in the cloud storage system. The Service Level Agreement (SLA) for cloud storage services includes a guarantee that the input/output operations per second (IOPS) will be at least as high as the user volume I/O IOPS, or input/output operations per second, 99.9% of the time. One of the main ideas behind our approach is that the IOPS metric should be taken into account whenever the scheduler is deciding where to put a new volume. The default scheduling mechanism is always keeping tabs on how much free and taken up space each host has. The new scheduling method keeps tabs on both the free and used IOPS counts. Updates to the Cinder scheduling algorithm in the reworked OpenStack should take into account not just the number of I/O operations per second (IOPS) a given host is capable of doing, but also the number of I/O operations per second (IOPS) a given volume can perform. When a storage node has available IOPS, it may allocate a new volume to use those resources. Now that I/O operations per second (IOPS) can be tracked, the volume allocation method is not limited to the amount of storage space available but also takes into account the volumes themselves and the services they provide. Consequently, cloud providers may provide greater I/O throughput SLAs to end users thanks to the enhanced scheduling algorithm's ability to acquire control over I/O performance. We bind the IOPS property during the weight sorting phase and look at the efficiency of the Cinder filter scheduler.

- Selecting a storage node with adequate free capacity to accommodate a volume's request is an example of available space filtering.

- Sorting I/O throughput by weight: based on the qualifying host list of free space, choose an appropriate available IOPS to meet the volume requirement. What's more, the node with the highest percentage of used storage space is the one that gets picked.

Available space filtering

Our selection of the most appropriate storage node from the available hosts is predicated on an existing space filtering method, and the host list is then subjected to a weighted sorting process in order to ensure compliance with the SLA. We illustrate storage node filtering with an example in Fig. 2. This has three storage nodes:

- Host 1. The first host has 600 GB of free space and 600 IOPS of available I/O throughput.

- Host 2. The second host, which has 800 GB of storage capacity and 400 IOPS of I/O bandwidth, is available.

- Host 3. Space on host 3 is 500 GB, and I/O performance is 550 IOPS.

The hosts in Experiment 1, Experiment 2, and Experiment 3 all have sufficient free space, therefore they remain on the list after being filtered when the volume request calls for the creation of a 300- GB volume with 500 IOPS. When a SLA violation occurs, cloud storage providers should install additional storage devices or storage nodes if the list is empty.

Quantifying the importance of inputs and outputs for filtering, the I/O throughput weight sorting algorithm uses the standard practice (available space scheduling).

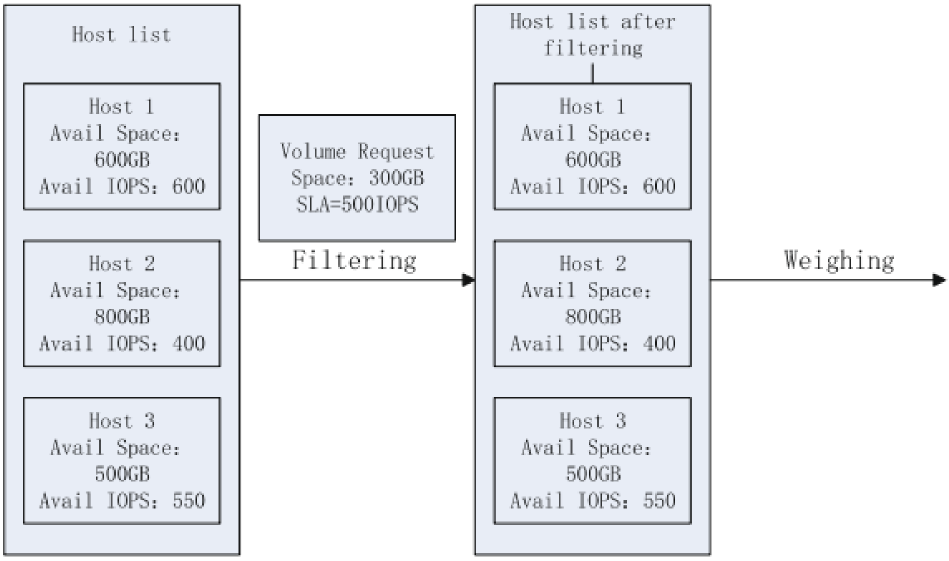

Fig. 2 Available Space Filtering

algorithm). In addition, the weight sorting process takes into account the I/O throughput characteristic. In Figure 3, you can see our algorithm's weight sorting in action. All of the data in Figs. Given that an enhanced weight coding scheme is based on the well-known 0 1 Knapsack Problem, solutions 2 and 3 are equivalent. Below is a breakdown of our scheme's weight sorting algorithm:

$$max \frac{S_v \cdot V_k}{S_k - A_{sk}} \cdot 100, \forall k \in K$$

$$V_k \leq T_k - A_{tk}, k \in K, V_k \in \{0, 1\}$$

If K is a set of storage nodes and V is a request for a new volume, V_k∈0, 1> indicates that if the volume is placed on storage node K, then $V_k = 1$, and otherwise $V_k = 0$. The symbol "S" stands for "space," "S_k" denotes the entire size of the storage node "K," and "S_v" stands for "volume request" storage space size. The IOPS throughput is denoted by T, the IOPS throughput of a single storage node, K, is denoted by Tk, and the IOPS throughput of a single volume request, Tv, is denoted by T_k+T_v. The storage node K's allotted space size, Ask, and its assigned IOPS throughput, Atk, are denoted as follows.

Two of the storage nodes, 1# and 3#, meet the requirements of the weighted sorting method described above. If we're looking for the best node in terms of storage space usage, then Host 3# is it, followed closely by Host 1#. It's possible that may not just make the accessible, but

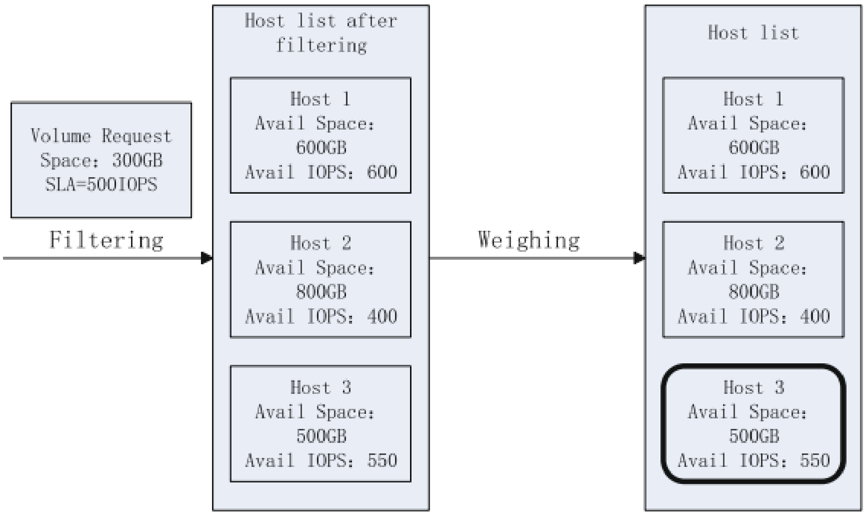

Figure3. I/O Throughput Weight Sorting

In addition to meeting SLA requirements, volume IOPS causes storage node space use to peak. If the host list is not empty and the aforementioned limited requirements are not met, the volume request will be queued until there is sufficient space. Furthermore, this indicates a breach of the Service Level Agreement. The addition of additional storage devices or storage hosts from cloud storage providers is required.

SECURITY-AS-A-SERVICE

In the eyes of the average user, security concerns may mean a wide variety of various things depending on the setting, the people, and the resources at play. Despite this, it is typical to speak about security issues together, with the goal of tackling every issue at once. Since security is the need that can scarcely be satisfied and monitored, the Cloud Computing method exacerbates the sense of security deficiency. This is because it brings together under the same roof an extraordinary number of diverse technologies and techniques. The identification and protection against vulnerabilities via the implementation of suitable security procedures is the foundation of most security solutions. The adoption of virtualization, of additional abstraction layers, and the flexibility of the infrastructure make vulnerability analysis very difficult, and thus the application of well-known security techniques (network security, protection against denial of services, and access control) is not sufficient. However, at the present time, there is no viable option accessible.

Providing Security as a Service is challenging since security mechanisms are often already built into infrastructures and extend to other services in unexpected ways. By comparing and contrasting the five qualities with the Security-as-a-Service model, we can highlight how the on-demand self-service implies that users should be able to activate security features without human interaction. This necessitates a system that can help people come to an agreement and negotiate terms automatically. In addition, security measures should be implemented from the perspective of the end user, rather than the provider's offerings. The open nature of the Internet and the need for simple configuration and activation of security measures necessitate the provision of security services in light of the widespread nature of network access. An emphasis on security from the user's point of view is needed once again. In addition, Security-as-a-Service must be dynamically added to preexisting services owing to the resource pooling feature. If several security measures are implemented, the same service must be provided in various forms. Rapid flexibility necessitates a simple re-configuration of an existing service to accommodate the activation of a new security service.

Finally, the capacity to quantify a service's characteristics implies the possibility of gauging the security degree actually provided to end users. Because there aren't any tried-and-true methods for quantifying security, measuring the quality of services provided in the cloud is a challenge as well. While the need for security as a service is real, SaaS itself is not yet the answer. Customers in the cloud would benefit from an easy way to implement additional layers of security for their existing services, but doing so would raise concerns about the level of intrusion into managed cloud infrastructures and the ability to assess the security of these services from the perspective of the customers who use them.

CLOUD COMPUTING SECURITY AND PRIVACY ISSUES

The central issue of this chapter is discussed here; namely, the difficulties associated with cloud computing security and privacy. Given that cloud computing involves so many different technologies (networks, databases, OSs, virtualization, resource scheduling, transaction management, load balancing, concurrency control, and memory management), it's no surprise that there are a lot of security concerns surrounding it. This means that the security concerns that have plagued these other systems and technologies also apply to cloud computing. For instance, a cloud's security relies on the integrity of the network that links its many components. There are also several security issues because of the virtualization concept used in cloud computing. One such precaution is the safe implementation of virtual machine mapping to actual hardware. For data to be secure, it must be encrypted and proper protocols must be in place to govern data sharing. Security is also required for methods used in memory management and resource allocation. Last but not least, data mining methods, as is often done in IDSs, may be useful to malware detection in the cloud (Sen & Sengupta, 2005; Sen et al., 2006b; Sen et al., 2008; Sen, 2010a; Sen, 2010b; Sen 2010c). Figure 4 from the Trusted Computing Group's White Paper (2010) illustrates the six components of the cloud computing environment that call for special security considerations in terms of both hardware and software. Data at rest security, data in transit security, user/application/process authentication, data isolation, cloud legal and regulatory concerns, cloud incident response, and incident response are the six areas that need to be addressed.

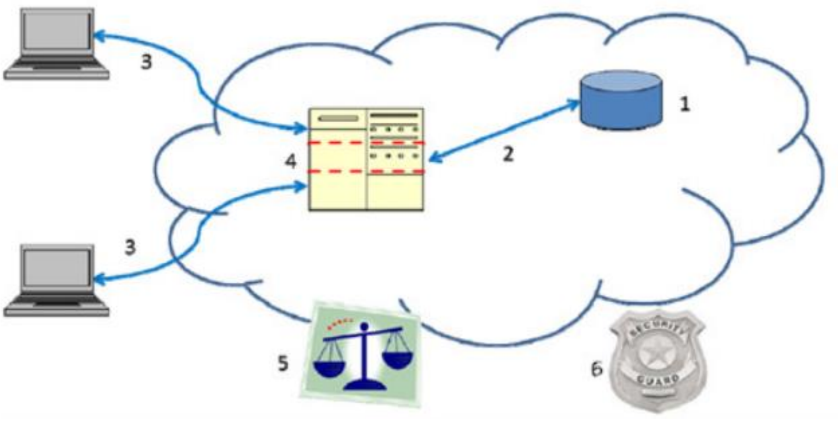

Figure4: Areas for security concerns in cloud computing: (1) data at rest, (2) data in transit, (3) authentication, (4) separation between customers, (5) cloud legal and regulatory issues and (6) incident response.

Cryptographic encryption algorithms are without question the gold standard for protecting data while it's resting. Manufacturers of hard drives are currently selling self-encrypting drives that adhere to the trusted storage criteria established by the trustworthy computing group (White Paper, 2010). Automatic encryption at low cost and with minimum performance hit is provided by these self-encrypting drives, which include encryption technology within the disk. Data may also be protected via software encryption, but this method is less secure and takes longer to implement since an attacker might potentially steal the encryption key from the system without the owner ever knowing it was compromised.

CONCLUSION

The primary advantage of SPECS is the opportunity to provide security assurance to Cloud End Users in terms of Cloud services, via the management of the life-cycle of agreed-upon security parameters included in Cloud Service Level Agreements. An authentication technique is described by which interacting clouds may authenticate each other on the fly. Service Level Agreements are becoming more important in the cloud ecosystem, a world of multi-stakeholder information and service providing. The SLA approach is used to guarantee that the service execution in the foreign cloud is in accordance with the agreed upon SLA parameters between the user and the provider, and a service matchmaking technique is proposed to select the best IoT service matching user requirements among multiple foreign clouds. The low price and high adaptability of cloud computing make it one of the most alluring technological developments of recent years. Despite the uptick in pace and enthusiasm, certain serious, long-standing worries about cloud computing are slowing things down and might end up undermining the concept of cloud computing as a new way to acquire IT. Many businesses that may benefit from cloud computing have yet to make the switch, and even the largest companies that have made the leap are often just entrusting less critical information to the cloud. The results of the experiments suggest that the SLA violation rate may be reduced by combining the weight algorithm with the standard OpenStack space filter.

REFERENCES

1. Halabi, Talal & Bellaiche, Martine. (2018). A broker-based framework for standardization and management of Cloud Security-SLAs. Computers & Security. 75. 10.1016/j.cose.2018.01.019.

2. Silva, Carlos Alberto & De Geus, Paulo. (2015). An approach to security-SLA in cloud computing environment. 10.1109/LATINCOM.2014.7041843.

3. Tomarchio, O., Calcaterra, D. &Modica, G.D. Cloud resource orchestration in the multi-cloud landscape: a systematic review of existing frameworks. J Cloud Comp 9, 49 (2020).

4. Kochovski, P.; Paš ˇcinski, U.; Stankovski, V; Ciglari ˇc, M. Pareto-Optimised Fog Storage Services with Novel Service-Level Agreement Specification. Appl. Sci. 2022, 12, 3308

5. Ficco, M. & Rak, Massimiliano. (2012). Intrusion Tolerance as a Service: A SLA-Based Solution. 2nd International Conference on Cloud Computing and Services Science (CLOSER 2012). 10.5220/0003941003750384

6. SAAD MUBEEN, Management of Service Level Agreements for Cloud Services in IoT: A Systematic Mapping Study, IEEE Access, VOLUME 6, 2018

7. R. Maeser, "Analyzing CSP Trustworthiness and Predicting Cloud Service Performance" in IEEE Open Journal of the Computer Society, vol. 1, no. 01, pp. 73-85, 2020.

8. Kochovski, P., Stankovski, V., Gec, S. *et al.* Smart Contracts for Service-Level Agreements in Edge-to-Cloud Computing. *J Grid Computing* **18,** 673–690 (2020).

9. Irfan, Muhammad & Hong, Zhu &Aimaier, Nueraimaiti& Li, Zhu. (2013). SLA (Service Level Agreement) Driven Orchestration Based New Methodology for Cloud Computing Services. Advanced Materials Research. 660. 196-201. 10.4028/www.scientific.net/AMR.660.196.

10. Lawrence, Andy &Djemame, Karim &Wäldrich, Oliver & Ziegler, Wolfgang &Zsigri, Csilla. (2010). Using Service Level Agreements for Optimising Cloud Infrastructure Services. 38-49. 10.1007/978-3-642-22760-8_4.

11. MuHAMMAD KAZIM, A Framework for Orchestrating Secure and Dynamic Access of IoT Services in Multi-Cloud Environments, IEEE ACCESS, VOLUME 6, 2018

12. Rohini H. Joshi, A Survey on Various Security Issues and Challenges to Secure Cloud Computing, International Journal of Innovative Research in Computer Science & Technology (IJIRCST) ISSN: 2347-5552, Volume-6, Issue-3, May 2018