Int. J. Inf. Secur. (2014) 13:113–170
DOI 10.1007/s10207-013-0208-7

SPECIAL ISSUE PAPER

Security issues in cloud environments: a survey

Diogo A. B. Fernandes · Liliana F. B. Soares · João V. Gomes · Mário M. Freire · Pedro R. M. Inácio

Published online: 28 September 2013

Abstract In the last few years, the appealing features of cloud computing have been fueling the integration of cloud environments in the industry, which has been consequently motivating the research on related technologies by both the industry and the academia. The possibility of paying-as-you-go mixed with an on-demand elastic operation is changing the enterprise computing model, shifting on-premises infrastructures to off-premises data centers, accessed over the Internet and managed by cloud hosting providers. Regardless of its advantages, the transition to this computing paradigm raises security concerns, which are the subject of several studies. Besides of the issues derived from Web technologies and the Internet, clouds introduce new issues that should be cleared out first in order to further allow the number of cloud deployments to increase. This paper surveys the works on cloud security issues, making a comprehensive review of the literature on the subject. It addresses several key topics, namely vulnerabilities, threats, and attacks, proposing a taxonomy for their classification. It also contains a thorough reviewof the main concepts concerning the security state of cloud environments and discusses several open research topics.

Keywords Clouds · Cloud computing · Issues · Security · Survey

D. A. B. Fernandes (✉) · L. F. B. Soares · J. V. Gomes · M. M. Freire · P. R. M. Inácio
Department of Computer Science, Instituto de Telecomunicações, University of Beira Interior, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal
e-mail: diogoabfernandes@gmail.com; dfernandes@penhas.di.ubi.pt

L. F. B. Soares
e-mail: lsoares@penhas.di.ubi.pt

J. V. Gomes
e-mail: jgomes@di.ubi.pt

M. M. Freire
e-mail: mario@di.ubi.pt

P. R. M. Inácio
e-mail: inacio@di.ubi.pt

1 Introduction

In their infancy, computers would fill large rooms with expensive electronic parts to produce little processing output, consuming as much power as several hundreds of modern computers. Nowadays, however, those rooms are being replaced by a multitude of processing units, storage hard drives, and network devices, serving any purpose. This multitude of computing and infrastructure nodes can be organized to form a distributed system that combines resources in an efficient manner, supporting highly demanding intensive tasks like scientific simulations.

Two of the most well-known paradigms for distributed systems are *clusters* and *grids*. While *clusters* are designed in a more coupling and homogeneous approach, *grids* dwell over large scattered and heterogeneous networks. *Clusters* tend to be more costly due to the expensive machinery used, such as parallel supercomputers with tens of thousands of off-the-shelf Central Processing Units (CPUs). Cheaper approaches use middleware to connect standalone resources, namely desktop computers. MPICH [177] is an example of such middleware. *Grids*, on the other hand, are most commonly deployed by using typical desktop and home computers as slave computation nodes, creating an overlay network upon the Internet, for instance. The Large Hadron Collider (LHC) computing grid of the CERN, the European Organization for Nuclear Research, is a good example. Nevertheless, this approach has increased management and task assignment complexity, and obstacles in collecting and gathering results.

Based on the paradigms outlined above [85], *cloud computing* has emerged roughly in the year 2008 as a new distributed computing paradigm with the purpose of achieving the long dreamed *computing as utility*, a term first invoked as early as 1965 by Corbató and Vyssotsky [60]. *Utility computing* refers to computational resources efficiently wrapped as services. *Cloud* environments mix up virtualization techniques in order to provide an efficient way of dispatching resources on the minute. This allows to deploy a pay-per-use business model, meaning that customers get to specifically choose whatever resources (e.g., CPUs, memory, bandwidth, security policies, platforms, and hardware load) they require, reducing costs by paying only for what is subscribed to. The definitions of cloud deployment and service delivery models, as well as of the essential characteristics of *clouds*, accepted by the community in the field, were discussed by the National Institute of Standards and Technology (NIST) in [185]. The deployment models include public, private, hybrid, and community clouds, and Virtual Private Clouds (VPCs). The service delivery models include the Infrastructure-as-a-Service (IaaS), the Platform-as-a-Service (PaaS), and the Software-as-a-Service (SaaS). Finally, the distinguishing characteristics for this technology are broad network access, rapid elasticity, measured service, on-demand self-service, and resource pooling.

Clouds are placed in large facilities that are specifically cooled and protected for the equipments and data they house, as clusters are. Such facilities have an umpteen number of servers that compute and store customers data, therein called data centers nowadays. As of 2012, Cisco expected to see global data center traffic quadruplicate over the next 5 years [55], whereas global cloud traffic will make up nearly two-thirds of the total data center traffic. This exemplifies where the Information Technologies (IT) industry is heading: toward a future dependent on *cloud computing*.

Although the *cloud* characteristics are well understood, especially from a business viewpoint, the security state of *cloud* environments is yet puzzling. Despite the growth in *cloud computing*, per se implying that many enterprises adopted the model, several security issues raise severe concerns for some. In fact, major payers might hold back [285], choosing to keep infrastructures on-premises rather than moving them to outsourced locations. The NIST finds security, interoperability, and portability as major barriers for a broader adoption [186]. Moreover, in 2009, the International Data Corporation (IDC), a market research and analysis firm, harvested opinions among company Chief Information Officers (CIOs) on the most concerning cloud issues [89]. The results clearly highlight the security topic as it ranked first with 87.5 % of the votes, 12.9 % more than the study of the previous year [88], in which security also led with 74.6 % of the votes. As a consequence to the risks involved, businesses hesitate to move their data to off-site *clouds*. Armbrust et al. [18] heard saying multiple times that *"my sensitive corporate data will never be in the cloud"*, supporting this mindset.

The field of *cloud computing* is actively researched in both the industry and the academia. In the midst of studies in the literature, a large part concerns security on *cloud*

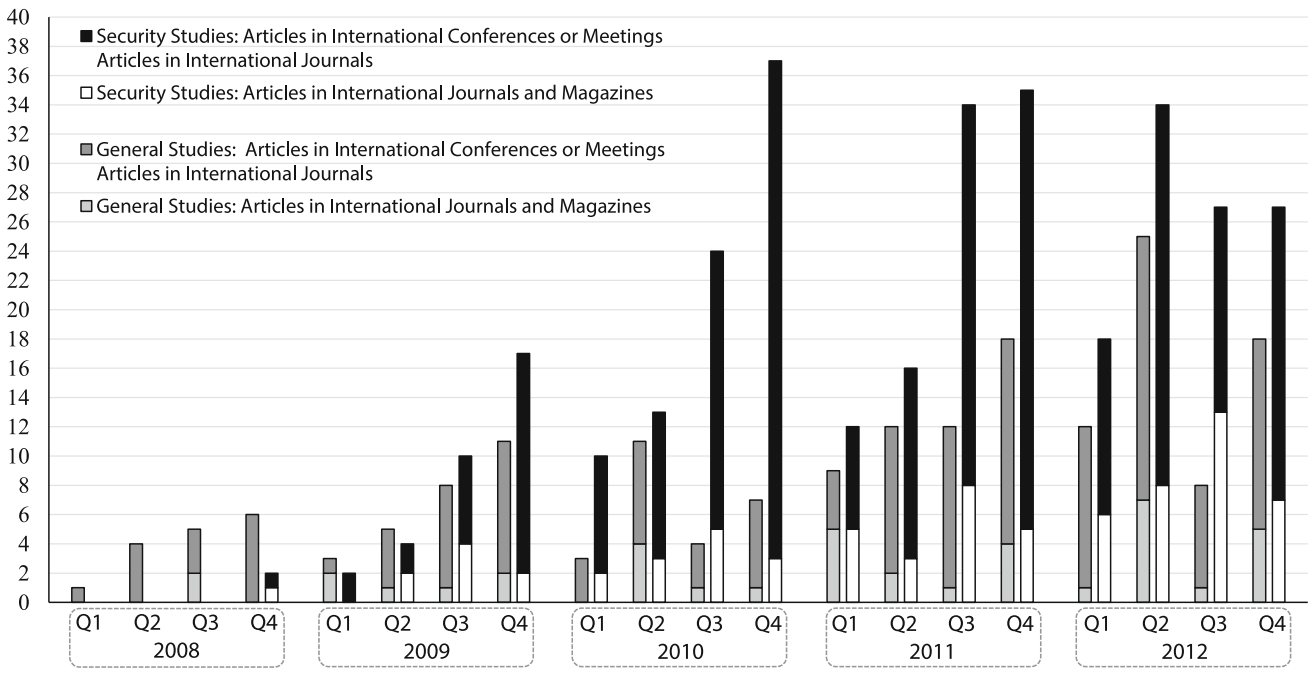

Fig. 1 Chart representing the number of papers on cloud-related topics found in digital scientific databases against the quarter in which they were published, from 2008 through 2012

environments, as shown in Fig. 1. This figure is the result of aggregating as many studies on the *cloud computing* topic as possible, including international conference, symposium, workshop, congress, and convention papers, as well as journal and magazine articles. Surveys and topic-specific articles were both considered. The results are divided into *General Studies*[1] and *Security Studies*, in order to emphasize the number of security-related studies. We were selective in this part of the work, choosing papers from well-ranked scientific journals and conferences or symposiums, all of them indexed by digital scientific databases such as the Association for Computing Machinery (ACM) Digital Library, Elsevier, the Institute of Electrical and Electronics Engineers (IEEE) Xplore, and Springer. Additionally, we used the discernment resulting from our revision work to filter out a few works that, in our opinion, were less interesting. We also excluded studies that are not specific to *cloud* scenarios, like the series of studies provided in [125,126,129], which look into service-based networks security. These studies have an indirect impact on the *cloud computing* paradigm but are not directly focused to them.

A total of 504 articles are included in the figure: 117 are journal and magazine articles, 387 are conference and other research meetings proceedings articles, and 12 are survey articles. Cloud general studies make up a total of 182 articles, while cloud security studies account to 322 articles. Even though we present only a portion of the number of studies published on the aforementioned digital scientific databases, we find it representative of the research trends in the field. In our opinion, the lack of interest in other investigation topics shows that researchers are concentrated in first mitigating security risks in *clouds* before exploring their wide area of potential applications. Thus, addressing the security issues in *cloud* environments seems to be of utmost importance to allow a better and more secure deployment of *clouds* throughout the industry. A clear distinction of those issues would help researchers with directions for future work. In addition, an overview of the security state would enlighten inexperienced newcomers to the field, raising their awareness on the topic as well. This provided the main motivation for this survey on *cloud* security issues.

The main contribution of this article is a comprehensive taxonomic survey on the cloud security topic, particularly on security issues. Unlike previous works, our effort is canalized to provide a more complete and thorough review of the research literature. The wide-scope analysis includes publications from both the industry and academia, and it describes several key notions of clouds in general and of enterprise security in particular. Those topics are introduced before entering the state-of-the-art discussion on cloud security issues. Throughout the text, the discussion focuses particularly on mentioning the classical security properties so as to identify the impact each issue may have. In addition, several real-life examples of security incidents are provided to better contextualize the discussion with the security landscape that the industry is facing. General studies are cited so as to contextualize the reader with the fundamentals of cloud computing or to help complementing certain ideas. Studies on cloud security issues are either cited as general studies, but within a discussion related with security, or linked to the security issues that are described in the respective subsection, may those be vulnerabilities, threats, or attacks. Furthermore, a taxonomy of security issues in cloud environments is provided in this article, clarifying to which extent cloud security spans. The analysis of the several topics covered in the survey provides the means to also discuss open research challenges and recommend future research directions on the subject at the end of this article.

The remainder of this article is organized as follows. Section 2 presents the related work and elaborates better on the contributions of this survey. Section 3 provides an overview over general characteristics of clouds and key concepts of cloud security. Subsequently, a discussion of the current published literature on the subject of cloud security issues is presented in Sect. 4, and a summary of that discussion and open challenges is included in Sect. 5. This article ends with the main conclusions in Sect. 6.

2 Related work

The security state has been and currently is widely discussed in both the industry and the academia. Several international conferences have focused on this subject alone, such as the ACM *Workshop on Cloud Computing Security*, the *International Conference on Cloud Security Management*, and the only European conference on the subject, *SecureCloud*, which already had three editions. Consequently, several scientific contributions have been published not only on conferences proceedings, but also in international journals. As such, several surveys on this area of knowledge have also been published, which are going to be described in this section.

Zhou et al. [310] elaborated a survey on the security and privacy concerns of many cloud computing providers. Security and privacy were discussed individually. While the first was studied with focus on availability, confidentiality, integrity, control, and auditing characteristics, the second was discussed by listing out-of-date privacy acts. In addition, a few problems related with multi-location storage were also discussed.

[1] *General studies* comprise studies not related with cloud security, such as mobile, scientific and green cloud computing, eGovernment, and optimization on cloud networks.

Vaquero et al. [284] provided deep insight into IaaS clouds security. The study focused on the security issues that multi-tenancy brings to cloud computing while analyzing them from the Cloud Security Alliance (CSA) point of view, that is, by categorizing security studies according to the CSA top threats to cloud computing published in 2010. Their work included describing security from the networking, virtualization, and physical sides of cloud IaaS networks.

Subashini and Kavitha [261] specifically studied the service delivery models security. After discussing the security in the scope of the several models, they analyzed them singularly, pointing out a greater number of issues in the SaaS model. An overview of current security solutions reported in the literature was also presented in that article.

Ahuja and Komathukattil [3] presented a survey on some common threats and associated risks to clouds. Approaches to tackle those threats and risks and security models of leading cloud providers were also presented.

Rodero-Merino et al. [232] have given a survey on the security state in PaaS cloud environments. They have focused on sharing-based platforms, focusing on the .NET and Java ones with emphasis on isolation, resource accounting and safe thread termination properties of the platforms.

Xiao and Xiao [300] provided a systematic review of security issues in clouds based on an attribute-driven methodology. The attributes used were confidentiality, integrity, availability, accountability, and privacy-preservability. For each attribute, a few threats were reviewed along with the corresponding defense solutions.

Aguiar et al. [2] wrote a book chapter focusing on the topics of computing and storage with regard to cloud computing security. The study overviewed several issues spanning various topics and recent developments regarding server storage and data computation security. Such topics include authentication and authorization, virtualization, Web services, accountability, and availability. Then, the discussion puts emphasis on techniques and mechanisms for achieving proper accounting, storage privacy, and public verifiability on outsourced data and computation.

Pearson [211] provided a comprehensive book chapter relating the privacy, security, and trust properties of cloud computing. The chapter introduces basic concepts, but focuses mainly on discussing the current security state of cloud systems. For that purpose, security issues and associated countermeasures are included in the work.

Pearce et al. [210] elaborated an extensive survey for the virtualization domain in a platform-independent manner, and particularly on the security problems around it. Their work first explains the basics of virtualization to then describe a broad architecture for system virtualization, with emphasis on network virtualization. The study discussed the incorrectness regarding assumptions of secure system isolation, oversight, and duplication and presented threats resulting from strong virtualization properties and from weak implementation of core virtualization requirements. Recommendations for securer virtualization implementations were also handed out.

Finally, Perez-Botero et al. [212] have provided a categorization of vulnerabilities on the Xen and Kernel-based Virtual Machine (KVM) hypervisors with basis on the open-source intelligence available in various vulnerability databases, including the National Vulnerability Database (NVD) and SecurityFocus. Their work focuses on three proposed fronts: the hypervisor functionality, the trigger source, and the attack target. Breakdowns for the vulnerabilities found are included in the article.

The security landscape concerning clouds is wide and the previous works focus on specific areas, paying less attention to the role that clouds play in IT and cybersecurity, though favoring sometimes the depth of the technical description of the solutions to the problems. Table 1 compares the several aforementioned works for different aspects, namely the topics they are focused in, the inclusion of industry references, the description of solutions to the problems, and the inclusion of a synthesis toward the end. Several symbols are used in the table to convey different meanings. For example, a ✓ is used to denote that a given aspect is covered in the article, while + or ++ are used to emphasize that particular attention is paid to a specific subject. On the other hand, a less detailed discussion on a given aspect is denoted by a –, while × is used to denote aspects not covered in the surveys.

The study presented herein differs from previous works for its broader scope. Rather than paying particular attention and detailing too much over the issues, a broader perspective of the state-of-the-art and high-level description is provided. Because of this, it is the only work proposing a taxonomy for the wide security landscape. This work also shows a concern in including pointers to real security incidents for each topic, which is not typically seen in other works. Furthermore, an analysis about the discussion of the security issues is provided at the end of the article, so as to deliver a series of guidelines and recommendations for future work and a discussion on an ideally secure cloud environment. This comprehensive study enables one to quickly catch-up basic concepts, review and understand the current security panorama of current cloud systems, analyze which security issues need to be addressed, and, consequently, identify opportunities for future research work. In addition, an analysis of the number and type of publications on the field throughout the years was presented in the previous section. For the sake of consistency, like in other works, the survey is complemented with key concepts of the cloud computing technology and its security state.

Table 1 Comparison of the related works with the survey presented herein regarding the security landscape, industry references, security incidents and issues, solutions, and summary effort

Survey	Year	Topics focused	Security landscape	Industry references	Security incidents	Security issues	Solut.	Summary
Zhou et al. [310]	2010	Industry technologies, legal problems, privacy acts	×	–	–	+	++	×
Vaquero et al. [284]	2011	IaaS clouds, networking, virtualization, physical	×	+	×	+	×	✓
Subashini and Kavitha [261]	2011	Software, Internet, Web, storage, access	×	–	×	++	+	×
Ahuja and Komathukattil [3]	2012	Software, perimeter, virtualization, compliance, access, storage	×	–	×	++	–	×
Rodero-Merino et al. [232]	2012	PaaS clouds, isolation, resource accounting, and safe thread termination	×	×	×	+	+	✓
Xiao and Xiao [300]	2013	Confidentiality, integrity, availability, accountability, privacy-preservability	×	×	×	++	++	✓
Aguiar et al. [2]	2013	Access, virtualization, availability, accountability, storage, computation	×	×	×	++	+	×
Pearson [211]	2013	Privacy, trust, legality, laws, compliance, access, storage, software, virtualization	++	++	×	++	+	×
Pearce et al. [210]	2013	IaaS clouds, virtualization, hypervisors, virtualized networking	×	+	×	+	+	✓
Perez-Botero et al. [212]	2013	IaaS clouds, hypervisors, vulnerabilities	×	+	×	+	+	✓
This survey	–	Several cloud-related security topics	✓	✓	✓	✓	✓	✓

3 Cloud security-related concepts

In this section, the fundamentals of the cloud computing model are presented. Whenever possible, the concepts are discussed while having their security context in mind. This section complements some of the ideas already discussed in Sect. 1 with the purpose of building a baseline for understanding the remaining part of this article.

3.1 Cloud service delivery models

Web 2.0 and cloud systems have given rise to a new class of services that captivate an increasingly connected population. In fact, according to Cisco, the IT industry is progressively moving to an Internet of Everything (IoE) [56]. The shift to cloud computing is a critical step toward that objective and, therefore, so are the service delivery models. Several studies introduce these [25,34,93,138,147,208,239,261,300]. The three delivery models are the IaaS, the PaaS, and the SaaS, sorted upwardly, and are illustrated along with the surrounding components in Fig. 2. In addition, the figure is complemented with some noteworthy security studies on the cloud stack. The operations of all models are supported by an IT-related infrastructure: the facilities that house the hardware, such as servers and network devices, and the cloud operating systems. Above the models, a network, such as the Internet, constitutes the intermediate layer—the medium—between clouds and customers. Transversely to the models, specific administration and business support strategies are employed to better manage the cloud and meet the customers needs. Trust extends itself throughout the stack as it is required to trust in infrastructures belonging to providers, except for the

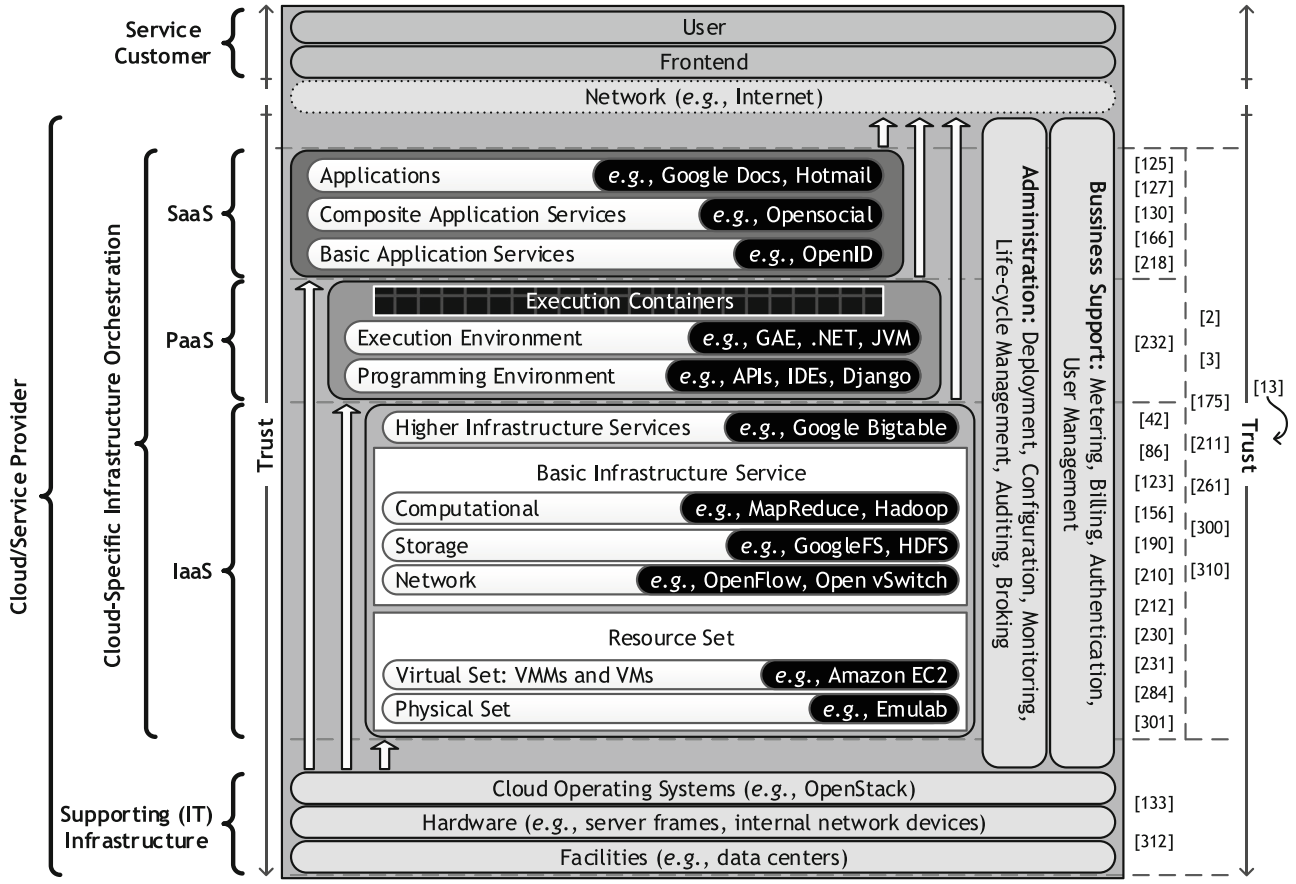

Fig. 2 Cloud service delivery models and inherent higher-level components. Examples and noteworthy studies are attached to each model, which are complemented in the figure by showing also the underlying IT infrastructure and top layer, which delivers the frontend and supports the interactions with the user (based on [100,147,211,300])

network layer because trust in the Internet is null. A discussion on each model is included below.

3.1.1 Infrastructure-as-a-Service

The bottom model, IaaS, revolutionized how businesses invest in IT infrastructures. IaaS providers, such as Elastic Compute Cloud (EC2) [10], offer Virtual Private Server (VPS) on the minute, paying only for what is needed. Rather than spending great amounts of funds on their own hardware and then hiring specialized technical crews to assemble the materials and manage them, this approach abstracts businesses from the management, provisioning, and scalability issues of the infrastructure, allowing them to focus on promoting their applications. This is achieved by elastically allocating physical or virtual resources on-demand, delivering storage, networking, or computational capabilities in the form of wrapped services, corroborating the utility computing side of clouds. IaaS provides basic security, including perimeter defenses, such as firewalls, Intrusion Prevention Systems (IPSes), and Intrusion Detection Systems (IDSes). Load balancing can also be included in this discussion as it is subjectively associated with availability attacks. Virtual Machine Monitors (VMMs) are critical components in cloud computing. They should provide complete isolation throughout all Virtual Machine (VM) instances. However, there are severe issues concerning this matter, discussed afterward. A cloud provider should, at least, ensure security up to the VMMs, which includes environmental, physical, and VM security.

3.1.2 Platform-as-a-Service

PaaS, the middleware model, allows customers to build their own applications by delivering services in the form of program development tools, platforms and frameworks—a container where customers run their components. Applications are then served by the upper model. The expenses on this model are also considerably lowered to companies, since they do not need to manage the hardware and software required to build applications. Google App Engine (GAE) [97], a PaaS provider, for instance, features Software Development

Kits (SDKs) for programming in Python, Java and Go. Apprenda [16] delivers solutions in .NET and Java also. Rodero-Merino et al. [232] enlightened of the fact that PaaS providers are twofold. There are clouds that share underlying resources (e.g., runtime components, libraries, and database engines) between tenants and others that do not, providing instead pre-packaged disk images with the software stack the customer demands. In the latter case, VMs provide the isolated system, although that may not be completely true in all cases [227]. Consequently, the PaaS model becomes more extensible than SaaS, providing a set of customer-ready features, delivering also greater flexibility on additional security. Clouds host Web Service-Oriented Architecture (SOA) applications that hide the underlying elements. Therefore, and because attackers are most likely to attack visible code, sets of security coding metrics should be put forth to quantify the quality of written code and avoid producing applications prone to attacks. PaaS customers do not have to worry about platform upgrades. All is managed by the PaaS provider. Despite the container provided by PaaS clouds, Rodero-Merino et al. [232] emphasized that such a layout can be compromised by malicious tenants in a straightforward way.

3.1.3 Software-as-a-Service

The top model, SaaS, allows applications to be remotely deployed and hosted in clouds, referring not to the means to create software as in PaaS, but a business model to distribute software. Subsequently, applications are accessed via the Internet, in turn constituting one of the major threats. This model improves operational efficiency and also reduces costs to customers by streamlining applications maintenance and support to providers. Without the need to install programs, a browser can be used to support user interaction with the applications. The SaaS model is rapidly becoming prevalent in the cloud business as it meets the requirements of IT companies. Yet, many security issues related with the building blocks of SaaS applications are known. Web is the technology of choice, making it the prevalent solution in the market for developing applications across the Internet. The existence of Web browsers that can incorporate many language processors, plugins, and addons makes them suitable to access a panoply of applications. However, vulnerabilities are discovered from time to time, which make way for malware proliferation. From the customer perspective, it is hard to understand whether or not data is well secured and applications are available at all times [49]. The difficulty lies on how to preserve or enhance security formerly provided by hosting systems [63]. More concerns arise in public clouds because specific pieces of data may be among other types of data completely unrelated.

3.1.4 Anything-as-a-Service

Although most authors consider the previous models separately, Armbrust et al. [17] considered IaaS and PaaS to be similar. They joined them together arguing that the gap between these models is not crisp enough yet. In addition to the three service delivery models, the literature describes one particular approach named Anything-as-a-Service (XaaS) [25,224], which refers to the fact that cloud systems are able to support and offer anything, or everything, in the form of services, ranging from large resources to personal, specific, and granular requirements. Examples include Data-as-a-Service (DaaS) [290], Routing-as-a-Service (RaaS) [43], and Security-as-a-Service (S_{ec}aaS) [5]. XaaS security analysis naturally depends on each context.

3.2 Cloud deployment models

Due to the great diversity on cloud solutions the industry is now offering, customers should first look into available cloud deployment models to analyze their advantages, disadvantages, and constraints in terms of scalability, elasticity, pricing, or migration, for example. Mainly, they should be assessed in terms of security. For that, five models are discussed throughout the literature [2,34,103,172,219,238,255,261,302]. They are *public*, *private*, *hybrid*, and *community* clouds, and another type less studied named *Virtual Private Cloud* (VPC). These models are summarily described in the following subsections, paying particular attention to their security aspects.

3.2.1 Public cloud

The infrastructure behind a public cloud is, in general, owned by a cloud provider. A public cloud houses many services from different customers, therefore being accessed from multiple locations by multiple tenants. Web interfaces are commonly used to access the services. This model is based on a pay-per-use business approach and is typically low cost, supplying highly scalable services. The resources of the cloud are located at an off-site location, which turns this model into less secure and more risky than other deployment models, because the service delivery models can be subjected to malicious activities. In this case, Service Level Agreements (SLAs) between customers and providers must be well detailed and analyzed.

3.2.2 Private cloud

A private cloud has a proprietary infrastructure and may be placed within the internal data center of an organization, usually behind a firewall. Thus, the management and security responsibilities are much easier to carry out and identify,

which may be in charge of the organization itself or of a third-party. In contrast, private clouds encompass big budgets and require highly skilled IT technicians to manage them and improve security, control, compliance, resiliency, and transparency. Off-premises private clouds are expected to grow in 2013, so as to overcome sharing issues and compliance requirements [167].

3.2.3 Hybrid cloud

A hybrid cloud is a mixture of two or more other cloud deployment models that are centrally managed and circumscribed by a secure network. It is traditionally seen as a mixture of private and public clouds, bringing together the advantages of each one and overcoming their obstacles. It allows multiple, but limited, and well-defined entities to access the cloud via the Internet in a more secure manner than public clouds. It also enables data and application portability. This model is managed by both the organization and a third-party entity and is placed in both on-site and off-site locations.

3.2.4 Community cloud

The community cloud deployment model is the one that is controlled and shared by multiple organizations. Usually, the cloud is setup to support a common interest among the several owners. It may be managed by the owners committee or a third-party organization and may be placed at an on-site or off-site location. The members of the community can freely access the data in the cloud. The community cloud eliminates the security risks of public clouds and the costs of private clouds.

3.2.5 Virtual private cloud

This last model is mentioned by less sources, and it consists on using Virtual Private Network (VPN) connectivity to create virtual private or semi-private clouds, resorting to secure pipes supplied by VPN technology and by assigning isolated resources to customers. A VPC seats on top of any model previously described, likewise a VPN that is built upon other networks. Hence, a VPC is a particular case of private cloud existing within any other. This model allows entities to use cloud services without worrying about operating in shared or public environments [121]. An example of this model is Amazon VPC [11].

Table 2 summarizes the main formerly discussed characteristics of each cloud deployment model. Even though there is no information characterizing ownership, management, location, and cost for VPCs in the literature, their characteristics are inherited from the underlying models due to already discussed facts. Each model presents its own problems and specific security issues. Businesses must take into account several factors, namely available budget, purpose of the cloud, and security requirements, before deciding on a specific model.

As emphasized by the NIST [186], interoperability between clouds is still a barrier that needs to be overcome. Although Cisco thinks the hybrid approach is the future of cloud computing [56], for now it is still confusing and unclear, because the rush to the cloud created a diversified cloud industry. Nebula One [181], for instance, a product of the Nebula company that dedicates to private clouds, is a sleek private cloud solution that acts much like a single computer, being easily turned on or off. The customer has the ability to choose the number of cores, combined storage, and memory of the product infrastructure, which provides Application Programming Interfaces (APIs) compatible with OpenStack and Amazon EC2 and Simple Storage Service (S3). The expectations for this solution are high [71]. The rapid growth culminated in a state almost devoid of standards and interoperable cloud networks. Thus, it is rather difficult or impossible to interconnect distinct clouds in a collaborative seamless fashion, a concept called *intercloud* [32,233]. The term refers to a network of clouds, a place of cloud computing, interoperability, ubiquitous and utility computing, and data storage. A well-founded infrastructure must exist to support *interclouds*, provided by, for example, topologies, standardized communication protocols, trust models, identity and access management, encryption and key management, and governance considerations. *Interclouds* would overcome the *lock-in* issue faced by customers and free data movement among distinct clouds.

Table 2 Summary of the main characteristics of the cloud deployment models, regarding Ownership (Organization (O), Third-Party (TP), or Both (B)), Management (O, TP, or B), Location (Off-site, On-site, or B), Cost (Low, Medium, or High), and Security (Low, Medium, or High)

Deployment model	Ownership	Management	Location	Cost	Security
Public	TP	TP	Off-site	Low	Low
Private	O or TP	O or TP	On-site	High	High
Community	O or TP	O or TP	On-site	High	High
Hybrid	B	B	B	Medium	Medium
VPC	B	B	B	Low	High

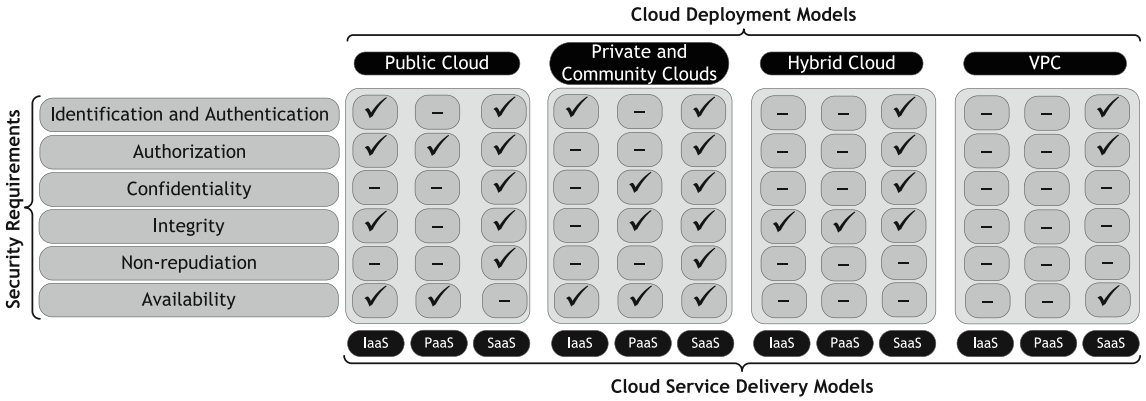

Fig. 3 Security requirements per cloud service delivery model and for the public, private and community, hybrid and VPC deployment models, as established by Ramgovind et al. [219]. A *check mark* means an obligatory requirement in the combination of a specific service delivery model with the underlying deployment model, whereas a *dash* means optional

3.3 Cloud deployment and service delivery models security requirements

This subsection complements the discussion of the cloud deployment models by introducing their security requirements per service delivery model. Businesses should conduct strategic evaluations of each model before choosing one of them. Figure 3 summarizes six security requirements: identification and authentication, authorization, confidentiality, integrity, non-repudiation, and availability. As can be seen on the figure, authorization requirements on IaaS, PaaS, and SaaS models on public clouds are mandatory to prevent unauthorized access to assets. The hybrid model requires less properties than the public and private models as it is more secure. Among the public, private and community, and hybrid deployment models, integrity is a very desired requirement, pointing out the interest in checking data correctness and whether it was tampered with or corrupted. This calls for auditability and integrity-checking mechanisms, such as the High-Availability and Integrity Layer (HAIL) [35]. Furthermore, requirements in the SaaS model span throughout all three deployment models, as corroborated by the survey provided by Subashini and Kavitha [261]. The majority of the requirements is in the SaaS model, which adds reasonable concerns to the Web- and service-based access of SaaS applications. The VPC is a less stringent model because a specific part of the cloud is allocated to one customer in an isolated manner. Obligatory requirements for VPCs are identification and authentication, and authorization for access purposes, and availability. The remaining are optional because customers have remote control over their cloud infrastructure, choosing which VMs they want to instantiate and which configurations apply for the underlying network and hosted applications.

3.4 Data center security

It was previously said that clouds resemble cluster systems, not only in coupling together computing resources while having a common goal, but also in rooms especially designed to cool and protect equipments. Data centers are thus built while having in mind many geological and environmental aspects, such as location, temperature, humidity, and earthquakes probability. Other aspects include political, governmental, and energy-saving aspects. With strong physical foundations (e.g., grid redundancy [54]), cloud providers assure that the cloud uptime is very high [50], reaching 99.99 %, and is fully fault-tolerant, thus achieving the tier four level in many cases. Tier levels are used to classify data centers quality, being the lowest level 1 and the highest level 4. The goal is to achieve highly reliable and available facilities in terms of uptime and elastic resources. In fact, cooling is also an active research field with many techniques available specifically designed to cool IT rooms.

Physical security is established on-site throughout a data center. Other security measures would be unnecessary if this prerequisite was not fulfilled. Data centers must be well secured (e.g., using a security center for managing video cameras and personnel entrances) in order to prevent break-ins and other physical violations. Access to the massive computation servers, storage servers, and network equipments should be physically restricted, allowing only exclusive personnel with security clearance to perform managing operations. In fact, private identity cards assigned to each employee are many times used as means to open door locks and access certain areas of the facilities. Providers might also lay further security options to customers, though with a higher price associated. For instance, racks might be surrounded by cages with padlocks, to which the opening keys are kept with the customers. In addition, a weighting chamber might

be installed before entering IT rooms so as to check the exit weight of the persons who entered. This approach is useful to find out whether any equipment was stolen inside.

The internal networks of cloud computing environments can be composed of service-driven networks, Storage Area Networks (SANs), and computational- and storage-related hardware. Hence, as any other enterprise network, perimeter security must be deployed to analyze network traffic and safeguard data in transit. Network security approaches include firewalls and IPSes to prevent security incidents; IDSes to alert malicious intrusion attempts [150,162]; and honeypots to create distractions for attackers and therein learn their movements [252]. Typically, a Security Operations Center (SOC) is established within the facility, monitoring and analyzing network health to detect pattern anomalies. A Computer Security Incident Response Team (CSIRT) placed within the SOC collaborates with other CSIRTs around the globe to share intelligence and aid in security incidents if necessary. Security Information and Event Management (SIEM) solutions are mandatory in order to obtain a high-level perspective of the network security status. SIEM solutions correlate real-time events triggered by perimeter defenses and security agents setup in each node within the network to learn what is normal and abnormal behavior. Hewlett-Packard (HP) ArcSight [114] is an example of a SIEM that performs event correlation. Security experts configure them in order to serve their alert requirements and purposes. Several SIEM platforms available in the market were compared by Kufel [142]. Various cloud IDS solutions are available nowadays [72,145,226]. Modi et al. [171] recommended IDS and IPS positioning in clouds to achieve the desired security in next generation networks, with particular attention to the trade-off between security and performance, as discussed by Patel et al. [206] in their state-of-the-art survey on IDS and IPS solutions.

Kant [133] conceptualized a four-layered model that subsumes modern data centers. The bottom layer is composed of the physical infrastructure, which aggregates server farms to form clusters. Then, a virtual infrastructure layer is built upon it. This layer enables to run co-resident VMs that can be setup to serve virtual data centers. A single virtual data center can be rented to a single customer, giving the customer full control over the management of VMs. The third layer is called a virtual infrastructure coordination layer, whose purpose is to tie up virtual data centers and cross-geographic location deployment. This layer mounts scattered virtual data centers, which can then be configured to build distributed virtual data centers. The last layer is for the service provider, which can be another entity involved in the cloud computing business or the very cloud provider. At the top of the model, applications run in a SaaS manner. Security matters should be regarded transversely to the whole model.

According to GigaOM, a media company, the data center infrastructure now extends beyond the four walls of the data center. A new realm of data centers is emerging. Nowadays, data centers are not just the machines, but are the data centers plus the network connecting them [70], further complicating the security requirements of clouds, and consequently of *interclouds*. For example, Google Spanner database, recently made public, syncs data across five data centers. Netflix, one of the biggest broadband traffic drivers, and Facebook also operate this way. Zissis and Lekkas [312] identified *flooding* attacks, *hardware interruption*, *theft* or *modification*, *infrastructure misuse*, and *natural disasters* as main issues to data center facilities. Note that the term *flooding* is related with the availability property when Denial of Service (DoS) states are achieved, therefore being part of a security requirement.

3.5 Cloud security reference model

The cloud security model depicts the actors in the cloud business and operation. It is composed of the cloud infrastructure and the entities that manage and ultimately use it. Cloud providers own data centers, having all the responsibilities regarding the management of the resources they contain. On the other hand, cloud customers and end users rent services from the cloud provider. An optional service provider can be included in the security model to represent the cases where cloud resources are rented to intermediate providers. This optional service provider is used in the model to enable the specification of what it is being rented. Additionally, SLAs are closed with providers so as to describe how services are executed and the terms of service. Typical SLAs include data exchange rates, mean time to repair, jitter and other service properties related with security as well [3]. While bandwidth, storage, or processing power are measurable parameters, security-related parameters are non-quantitative properties, thus comprising an obstacle.

The cloud security model described so far is schematized in Fig. 4, where it is possible to discriminate possible attack vectors. Dashed circles represent users that have closed SLAs with a service provider. In the model, two service providers are illustrated: one SaaS provider and one IaaS provider. Each provider is now able to sell services to end users. Also depicted, one supposedly normal user can turn aggro and act maliciously without apparent suspicion, being more stealthier than others. In addition, a malicious employee with privileged access and knowledge of the cloud resources can do considerable damage. Finally, across the Internet, a potential dangerous community can scan for vulnerabilities and exploit them afterward. Other ways to get inside the cloud network include getting access to login credentials of honest customers. Each actor, good or bad, can have more or less knowledge of the cloud and can produce more or less impact,

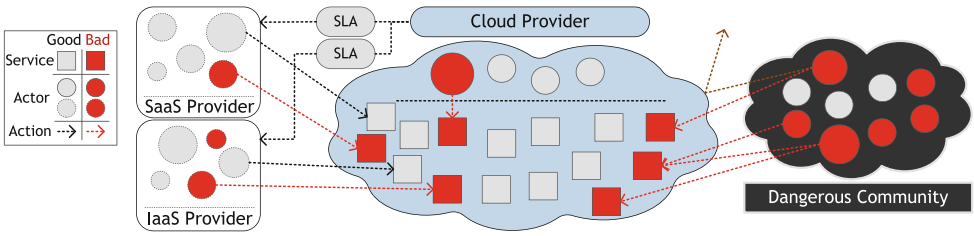

Fig. 4 Illustration of the cloud security reference model. Cloud stakeholders, SLA elements and interactions between one another are identified

and be more bolder, hence the different circles and sizes used for actors in the figure.

A noteworthy aspect is that, while cloud customers are responsible for application-level security, providers are delegated with physical and logical security responsibilities. Responsibility over problems on intermediate layers of the cloud stack is shared between the two entities. Cloud customers may, nonetheless, outsource their security responsibilities to third-parties who sell security-related services.

3.6 Important concepts in cloud security

Cloud security covers numerous subjects. In order to understand them, the underlying concepts that might identify the source of vulnerabilities and threats must be introduced. This subsection analyzes those concepts, starting with an explanation on virtualization elements and then on multi-tenancy. Cloud software is also discussed, followed by the discussion of the concept of data outsourcing. Then, data storage security and standardization are reviewed, and the section ends with a discussion on trust.

3.6.1 Virtualization elements

Virtualization consists in the process of abstracting computer applications, services ,and Operating Systems (OSes) from the hardware on which they run [252]. Typically, virtualization components include VMs and VMMs (also known as hypervisors). A VM image is a large-sized file of a prebuilt copy of the memory and storage contents of a particular VM, and the virtualized OS in it, called guest OS. The guest OS functions normally like a host OS, having multiple applications running on top of it, but with the difference that direct access to hardware is not provided. This access is mediated by the VMMs, which can allocate virtual hardware resources for each VM. Those resources include CPUs, memory, network adapters, hard disks, and others. If a new VM request is received by VMMs, a new instance is quickly created and resources are conveniently designated according to the request details. VMMs can create a virtual network to interconnect VMs [271,283]. To this end, VMs are linked to virtual switches and can be mounted to emulate external and internal networks, including DeMilitarized Zones (DMZs). In addition, specific VMs or virtual Network Interface Cards (NICs) can be linked to specific hardware NICs. VMware vSphere [286] supports such virtual features. Thus, VMMs control the creation and deletion of VMs, supporting the on-demand and elastic business model of cloud computing. Popular free VMM solutions include VMware Player [287], Oracle VirtualBox [197], RedHat-maintained KVM [222], Microsoft Hyper-V [169], and Xen [275], a project of The Linux Foundation. Popular commercial paid VMMs include VMware Workstation and vShpere [287], Oracle VM Server [204], Parallels Desktop and Virtuozzo [205], and Citrix XenServer [58]. While the former free solutions are usually more deployed for endpoint test usage, the paid solutions aim for production cloud environments, except for Xen and Hyper-V. KVM and Xen are underlined for their open-source approaches, being the latter the open-source version of XenServer.

Since VM images can be easily copied, moved or cloned to other locations, clouds can deliver highly available and scalable services. In case of having a machine compromised, or with lack of resources, or if it suffers an outage, VMs can be moved to other servers while keeping the integrity of their contents. Nonetheless, such functionality requires specific middleware, part of the VMMs and cloud OSes. Such virtualization techniques bring benefits like costs and downtime reduction, ease of management and administration, workload distribution and scalability [41,311].

3.6.2 Multi-tenancy

Multi-tenancy refers to the feature of being capable of running multiple instances under the same shared platform. Each instance can be accessed by one or more users, called tenants, while sharing a common platform. In an IaaS cloud provider, the multi-tenancy sharing platform refers to the VMM, while instances refer to VMs. In a PaaS provider, however, multi-tenancy refers to a Virtual Platform (VP) that can run multiple applications, such as .NET and the Java Virtual Machine (JVM) [232]. Nevertheless, because customers data may be stored at the same physical location, the multi-tenancy feature can be exploited in the form of *co-location*, *co-residence*, or *co-tenancy* attacks. These consist in somehow gaining access to neighbor VMs or running applications. Other issues incur, like DoS that can be achieved by consuming as much resources of the underlying shared platform as possible.

3.6.3 Cloud platforms

By definition, moving to the cloud implies outsourcing IT infrastructures. The customers does not have control over the off-site servers, and thus, some kind of workable frame is required in order to deploy business applications or services. In the case of a IaaS cloud provider, the underlying platform is a VMM—a virtualization layer. In the case of a PaaS cloud provider, such frames are delivered in the form of development platforms. These provide the tools required to build SaaS applications. Just like any other local application, APIs and Integrated Development Environments (IDEs) are properly provided, which depend on the underlying VP and, consequently, on the programming languages.

3.6.4 Data outsourcing

Nowadays, the industries widely use the outsource business model. It is the process on which responsibilities over certain subjects are delegated to contracted third-party services, usually another company. This favors both the capital expenditure (CapEx) and operational expenditure (OpEx) of customers. Data outsourcing takes this concept into the IT industry, delegating the duties of storage, computing and security to third-party off-premises infrastructures, owned and managed in a data center. However, the most important aspect about data outsourcing is that it establishes physical separation between customers and their data [279,300]. Customers lose control on their data, trusting those off-premises infrastructures and cloud providers. To overcome this problem, providers must guarantee secure data computing and storage.

3.6.5 Data storage security and standardization

Although classical cryptography can be applied in many computing scenarios, the cloud paradigm requires data to be remotely processed in plaintext. Not just that, integrity-checking techniques, authentication mechanisms to control data access and secure protocols throughout the Open Systems Interconnection (OSI) model layers should be deployed. Companies strive for obtaining high-level certifications like the International Organization for Standardization (ISO) 20000 and 27001, giving customers reassurance of the contract veracity. However, the shift to cloud computing brought certain difficulties in that field. Applying common techniques might not suit the cloud operation as data centers usually hold massive amounts of data for processing. It may be impractical to, for example, hash entire datasets, otherwise one would have to bear great *computational* and *communication overheads* [300]. Additionally, reliable data storage also implies backing it up every now and then. Clouds from the same provider can be spread through several data centers. This enables providing geographic redundancy to data, meaning that a copy of this data is migrated to another data center in order to avoid single point of failure. However, this can bring legal issues as later discussed in this article. According to Leopando [148], an accepted rule for backup is the so-called 3-2-1 rule. Since it is easy to copy data on the digital world, the rule consists in having at least three copies in two different formats with one copy off-site. In the cloud context, this rule could be applied by having two copies with the cloud provider and one on enterprise premises. The development of *interclouds* and standards would definitely help achieving the desired certifications and, consequently, a better cloud health.

3.6.6 Trust

Trust is a subjective measurable scale that can thrust decisions based on the beliefs of the decisions [262]. Evaluating trust is a multi-faceted and multi-phased phenomenon based on several factors that constrain a certain decision. Therefore, it is highly volatile and strongly depends on the underlying context. On cloud environments, trust issues arise because a customer infrastructure is located at a off-site foundation and is managed by a second- or third-party entity. These two factors imply a human factor not known to customers to interact with the infrastructure. Configurations of the underlying SaaS, PaaS, or IaaS infrastructure makes part of the responsibilities of the cloud provider. More importantly, this includes security management. In addition, trust refers to the infrastructures themselves, the bare metal, the hardware and the data centers. When potential high-value data is put in almost total dependence of someone else, questions arise, spanning from the smallest asset to the biggest security picture.

Yasinsac and Irvine [303] discussed trustworthy systems. These systems should perform as expected even under atypical conditions, may those be operational errors, human interaction, or hostile disruption. This implies trustworthiness to combine reliability, which refers to system performance when all parties cooperate with security. In turn, security refers to system performance when some parties are malicious. As Yasinsac and Irvine argue, the difference between trustworthy systems and the classical security perspective is that the former works toward advancing the organizational mission using security discovery. In other words, trust-based systems balance security with other activities in order to ensure the continuity of an organization and achievement of the underlying objectives.

As defined above, trust is a bit of an abstract concept that measures decisions. Hence, trust is not just about the infrastructures decisions, but also about the human element in the context of information security [276]. Humans are the edge, truly. If they were not, we would not be seeing malware continuing to proliferate across several industries. Ironically

or not, the technology sector is the most attacked one [84]. Thus, the mixture of people-to-machine, people-to-people, and machine-to-machine interactions matters in any given IT context, whether it is within an enterprise infrastructure or within a cloud system.

3.7 Taxonomy for cloud security issues

Before presenting the taxonomy for cloud security issues, a brief introduction to the concept of security issue is given, so as to better elucidate when the various security terms are invoked throughout this article. A security issue is a general term to address something—like an event or action, a software or hardware misconfiguration, or an application loophole—that is not as it supposedly should be in the context of security. The security community traditionally uses the terms vulnerability, threat, attack, and risk to further specify what the issue is, therefore being important to understand their differences [69]. So, vulnerability, or gap, is a flaw or weakness of a system, which can be compromised by a threat. The risk is the likelihood of a threat agent taking advantage of a vulnerability, in the form of an attack, and corresponding business impact.

Grobauer et al. [100] clearly distinguished the difference between *cloud-specific* issues and general issues. Their study, which is based on sound definitions of risk factors and cloud computing, states that cloud-specific issues must be intrinsic or prevalent in a core technology; have their root cause in the essential characteristics proposed by the NIST; are caused when tried-and-tested security controls are difficult or impossible to implement; or are prevalent in established state-of-the-art cloud offerings. Zissis and Lekas [312] categorized cloud computing threats into multi-tenancy issues, account control, malicious insiders, management console security, and data control. Sengupta et al. [248] discussed issues of four categories. The first category is *cloud infrastructure, platform, and hosted code*. The second category is *data*, while the third is *access*. Finally, the fourth category is named *compliance*. Aguiar et al. [2] did not explicitly provide a taxonomy for cloud security issues, but those authors divide their work into six categories: *authentication and authorization*, *virtualization*, *availability*, *accountability*, *storage*, and *computation*.

Former studies, however, lack the higher-level perspective of the security factors that affect cloud environments because they were also more focused. The taxonomy proposed in this article revolves around eight main categories: *software*, *storage and computing*, *virtualization*, *Internet and services*, *network*, *access*, *trust*, and *compliance and legality*. An illustration of the taxonomy is presented in Fig. 5. The figure allows the extraction of a mental picture of the security state in cloud environments and the identification of possible factors causing the cloud security fuzz. To the best of our knowledge, the taxonomy proposed in this article is the first attempt toward that objective. It helps to understand how far the security state in cloud environments stretches to. The taxonomy covers the issues present in the cloud service delivery models, meaning that storage and computing

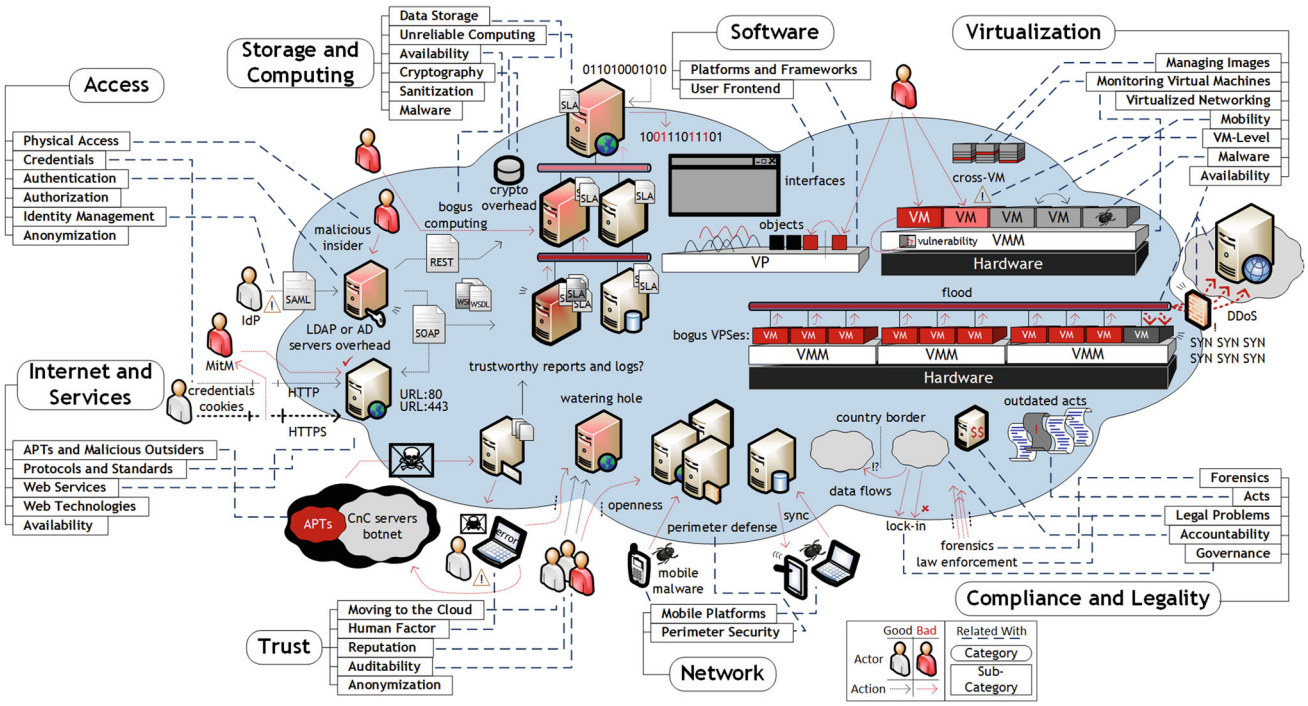

Fig. 5 Illustration of the taxonomy proposed in this article, showing the eight main categories and the several sub-categories

issues, virtualization issues, platform and software issues are included. Additionally, issues ranging from the Internet to the cloud-enabled enterprise network, to the very front door of clouds are also included. This drill down allows to better understand the attack vectors existent for cloud systems. Finally, two more areas of security issues are included, which are more subjective than others because trust, compliance and legal problems may not be directly related with the technology deployed in the majority of the cases. Each category is divided into some sub-categories that further address specific issues. This structure other sub-categories to be added in the future, if necessary. In the figure, some issues are related with some of those sub-categories so as to better understand what the discussions in the next section are referring to. Note that the categories and sub-categories were chosen while not having in mind where they fall within the cloud service delivery models, but which security issues are included in each one. The categories were chosen so as to minimize overlap (in terms of having issues falling into more than one category), while covering all possible security issues that may affect clouds. Additionally, the order of the categories in the figure is not the same as herein presented. In the following section, the assessment of the state-of-the-art security issues in cloud environments is done with basis on the taxonomy, following the structure and order previously presented in the text. Naturally, only cloud-specific issues are discussed in this article.

4 State-of-the-art on cloud security issues

Nowadays, cyberwarfare is a very complicated phenomenon to deal with. Interpreting it fully is not an easy task as state-sponsored attacks are more and more common to see, but nonetheless are very restrict. Very little information is publicly disclosed. Despite some skeptical people thinking that groups like Anonymous do not pose a threat to governments, history has proved that enterprises might not survive or sustain against one cyberattack.

The intense growth of cloud environments in the industry demands that new solutions must be devised. Faulty cloud implementations exist, and because of the large number of security issues discovered throughout the time, the move to the clouds may prove difficult for some. New approaches are, therefore, required to avoid being targeted and provide the leap to reach the next cloud frontier. This section discusses the security state of cloud environments thoroughly by describing its security issues. Except for the first subsection, each subsection of this section represents a category of the taxonomy proposed in this article. Each subsection is further branched to some topics that group security issues common in some property.

In the end of each subsection, a summary of the security issues discussed therein is included in tabular form. Such summaries focus on extracting the terms of the issues and agglomerating them according to the taxonomy proposed in this article. The issues and the works identified in the tables are mostly ordered according to the textual descriptions, so as to enable one to easily find the discussion on each one of the topics. Issues are, nevertheless, grouped per study, whenever applicable on a certain sub-category. The table not only includes studies of the academia, but also research works of the industry and a few articles from the social media, which are all discussed in the respective section. A dash (–) in the studies column of those tables means the issue was introduced by the authors of this article with basis on experience or on the study of the issues.

4.1 Industry research

Interesting research coming from industry has been published throughout the years concerning the IT security state. Vendors conduct their own research based on the data collected from their customers, periodically reporting findings about trends and evolution of threats. In the case of cloud computing, various studies have been published. Other more recent and general studies discuss IT in a wide-scope manner, but cannot dodge the cloud topic, also including interesting facts about it. Such studies aim at not only sharing intelligence with other security organizations, but also with the research community. This subsection covers some of those recent works along with pioneering works on the subject.

In 2008, the Gartner, a research and advisory company, published the Assessing the Security Risks of Cloud Computing report [87]. In this report, seven security risks were discussed from a customer viewpoint, clearly stating that such risks should be assessed before committing to any cloud solution. Those early risks were privileged user access, regulatory compliance, data location, data segregation, recovery, investigative support, and long-term viability.

In 2009, the European Network and Information Security Agency (ENISA), a security incident response agency for the European Union, published the Cloud Computing: Benefits, Risks and Recommendations for Information Security report [82]. Customer-related security risks were also enumerated on the document, listing loss of governance, lock-in, isolation failure, compliance risks, management interface compromise, data protection, insecure of incomplete data deletion, and malicious insider as top risks.

In 2010, the CSA, a nonprofit industry group dedicated to promote the use of best practices in cloud computing, provided version 1 of the Top Threats to Cloud Computing report [62]. The study spreads the security awareness in cloud environments with a few noteworthy publications focusing on it [138,174,281,284]. The report described the most popular threats to cloud computing and provided examples along with remediation directions for each threat. In the following year, the CSA published version 3 of the

Table 3 Top threats to cloud computing in 2013 as described by the CSA [64], the domains in which they are included, and the service delivery models they affect

Threat #	Name	Domain(s) #	IaaS	PaaS	SaaS
1	Data breaches	5, 10, 12, 13	✓	✓	✓
2	Data loss	5, 10, 12, 13	✓	✓	✓
3	Account of service traffic hijacking	2, 5, 7, 9, 11, 12	✓	✓	✓
4	Insecure interfaces and APIs	5, 6, 9, 10, 11, 12	✓	✓	✓
5	DoS	8, 9, 10, 13, 14	✓	✓	✓
6	Malicious insiders	2, 5, 11, 12	✓	✓	✓
7	Abuse of cloud services	2, 9	✓	✓	×
8	Insufficient due diligence	2, 3, 8, 9	✓	✓	✓
9	Shared technology vulnerabilities	1, 5, 11, 12, 13	✓	✓	✓

A *check mark* means the threat affects the underlying model. A *cross* means otherwise

report entitled as Security Guidance for Critical Areas of Focus in Cloud Computing [63]. In this study, fourteen domains of concern in cloud networks are identified. The first report contains references for each threat to the domains of the latter report, which are cloud computing architectural framework (domain #1), governance and enterprise risk management (domain #2), legal and electronic discovery (domain #3), compliance and audit (domain #4), information lifecycle management (domain #5), portability and interoperability (domain #6), traditional security, business continuity and disaster recovery (domain #7), data center operations (domain #8), incident response (domain #9), application security (domain #10), encryption and key management (domain #11), identity and access management (domain #12), virtualization (domain #13), and the new S_{ec}aaS (domain #14). Both studies are a major effort in reducing the security gap in clouds. The CSA further published an evolution of these works called The Notorious Nine Cloud Computing Top Threats in 2013 [64]. This latest report contains an updated list of the top threats, which grew in comparison with the list of 2010. The CSA top nine threats for 2013 are summarized in Table 3.

Although the majority of the studies point out that cloud security is dramatically lower that in other IT systems, the State of Cloud Security Report [6] allays such thoughts. The report has been published by Alert Logic, a company dedicated to security expertise. The data collected from its 1801 customers agglomerated up to one billion security events and were automatically analyzed and correlated by one of its security platforms. In the midst of those events, more than 45,000 incidents were observed between April 1 and September 30, 2012. Surprisingly, their main finding was that of cloud environments not being inherently less safer than enterprise data center environments. Moreover, cloud attacks tend to be more opportunistic crimes, whereas attacks to enterprise data centers are sophisticated and targeted. The prime example in the latter case is *spear-phishing*.

The 2013 Cisco Annual Security Report [56] discussed the wide panorama of current IT facts under the assumption of the IoE—an any-to-any world. The report gives insight into the heterogeneity of devices and how they changed enterprise models [e.g., Bring Your Own Device (BYOD)], endpoint proliferation, big data, malware and spam trends, and evolutionary threats (i.e., the combination of old attacks with new techniques). Notwithstanding, the cloud computing paradigm is also discussed. VMMs security is discussed along with the growth in quick, cheap, and easily available VPSes that can be used for criminal activities. The report further enlightens on virtual workloads and possible high-value data along with applications that move around the data center. It is stated that security must be a programmable element seamlessly integrated into the data center fabric.

The NIST has contributed to the field with its cloud computing reference architecture template in 2011 [184]. But, it has now provided a new addition entirely focused on security in its special publication entitled Cloud Computing Security Reference Architecture [187]. The document aims at demystifying the process of selecting cloud computing services that best meet the needs of customers in the most secure and efficient manner.

Table 4 summarizes the research works from the industry that partially or entirely overviewed the cloud security topic. The initial works were pioneering on the subject, serving as a baseline for accelerating the rate at which cloud com-

Table 4 Summary of the industry research works on the cloud security field

Report	Enterprise	Year	Pioneer. Work	Cloud-specific
[87]	Gartner	2008	✓	✓
[82]	ENISA	2009	✓	✓
[62]	CSA	2010	✓	✓
[63]	CSA	2011	✓	✓
[64]	CSA	2013	✓	✓
[6]	Alert Logic	2013	×	✓
[56]	Cisco	2013	×	×
[187]	NIST	2013	✓	✓

puting was better wrapped up and ultimately distinguished from other IT alternatives. Security was already a concern on those initial works. In more recent reports, the state of security in general is discussed with the cloud computing topic included. Both the NIST and CSA are highlighted for their major contributions in this area of knowledge.

The following subsections perform an extensive review of the research literature on cloud security issues. Vulnerabilities, threats, and attacks are discussed throughout the text with the respective studies. The review includes topic-specific issues that can relate specific cloud deployment or service delivery models, thereupon complementing previous discussions. The discussion follows the taxonomy proposed in Sect. 3.7.

4.2 Software security issues

Software security is, and has been for a while, a vital topic regarding computer systems. Nowadays, security measures might be hard to enforce because common software usually has thousands or millions of lines of code. To make it worse, that software can be written by several people with different programming skills and ideals. Even if all follow a set of pre-specified metrics to develop the software, a single bug can pose a critical problem. In critical and real-time systems, like the ones in airplanes, it is imperative to have fully reliable software that has passed rigorous software tests so that it does not fail because people lives are at stake in this case. Data, after an extract process, can be transformed into information. A business secret stored in a digital file is, therefore, a high-value piece of information. Although there are no lives at stake here, the enterprise revenue can be. Thus, cloud SaaS systems should ensure no data leakage by means of software faults. In spite of being in a more tightly managed environment, software is no more secure simply by virtue of being in a virtualized environment [210]. The following subsections discuss *platforms and frameworks* and *user frontend*.

4.2.1 Platforms and frameworks

Rodero-Merino et al. [232] provided an in-depth study on PaaS sharing-based cloud development and running platforms. Their study was focused on analyzing the security state of Java and .NET platforms in the multi-tenant context. Three topics were studied in each platform: *isolation*, *resource accounting*, and *safe thread termination*. Given that PaaS tenants can share platform resources, it is important to discuss what kind of isolation security such platforms provide, along with resource accounting and thread termination because, for PaaS providers, it is essential to comply with such properties in order to align them with the pay-per-use business model in a secure manner.

Java implements sandboxing for isolating running programs, bytecode for checking runtime integrity, and cryptographic and secure communications APIs. It also implements control over which classes can be instantiated by threads, by means of a class loader. The most straightforward way for guaranteeing isolation is to create one JVM per application. The drawback is that of expensive resource usage, mainly memory. Although not secure, another way for providing isolation is by using standard Java capabilities—a security manager that controls one class loader per application. This approach does not prevent leaked references and thread termination. Nevertheless, research has been put onward for providing secure Java isolation. Rodero-Merino et al. explain the Multitasking Virtual Machine (MVM) [67], isolates-based KaffeOS [22] and I-JVM [91] solutions, and a heap-based protection [263]. Isolated components are assigned to each application, giving them the illusion of executing in a non-shared VP. The .NET Common Language Runtime (CLR) provides a more secure isolation, by using the concept of application domain, which are isolated from code of other application domains. In terms of resource accounting, neither Java nor .NET provides capabilities for resource accounting. A generic API is provided by the MVM. Additionally, none of them can enforce termination of threads. The underlying methods to terminate threads for Java and .NET can be easily bypassed by handling exception catches. Both methods trigger exceptions to stop threads, but cannot force them to terminate. Additionally, in the Java case, terminating a thread can leave behind objects in an inconsistent state.

The studies pointed out by Rodero-Merino et al. [232], are, in their majority, prior to the rise of cloud environments. Although they natively address some issues, and can be extended to address others MVM seems the more complete solution. Access control mechanisms, reference leak, shared static references, block by synchronized static components, thread termination and resource accounting are all addressed by MVM. Therefore, it is expected to see Java being more adopted than .NET by PaaS providers. CloudBees [59] is one example.

What was discussed above is a responsibility of the PaaS or SaaS providers. Moreover, *unsafe* APIs and IDEs tied to a specific VP can render faulty or vulnerable code. *Insecure system calls* or *deficient memory isolation* [175], as seen above, are examples of *unsafe* platforms that may allow malicious binaries to run [29]. Bad Software Development Life Cycle (SDLC) approaches can have the same result as the aforementioned ones [161], but with the difference that, in this case, the responsibility is on the customer side. Therefore, a PaaS or SaaS provider must ensure that even bad code built by customers does not affect in any way the underlying VP.

4.2.2 User frontend

In Amazon Web Services (AWS), for instance, a customer can rent IaaS services through a Web interface available through the Internet. Afterward, a user interface with fine-grained configuration capabilities is provided to manage, orchestrate, and monitor the activity of the service usage [3,100]. On typical administrative interfaces internal to an enterprise, only a handful of privileged administrators has strict access to them. On a cloud environment, however, it is exposed to the Internet. The interface, by default, is a gateway into the cloud and makes it an attractive attack target that can compromise the overall service security [3,281], therefore requiring proper security measures.

Grobauer et al. [100] stated that there is a higher probability of *deficient configurations* and *unauthorized access* on such interfaces, because each customer has its own. Subashini and Kavitha [261] also stated that hacking through *application loopholes* or *injecting masked code* into an SaaS system can break isolation barriers (e.g., like the containers discussed in Sect. 3.1.2) put in place by VPs. Pearson [211] further said there is an increased risk of intrusion, even if access is controlled with a password. In addition, frontend interfaces are also deployed for administrators to manage VMs. VMMs normally have management consoles, such as XenCenter for Xen VMs. Such consoles, which can be accessed remotely, also bring up the vulnerable possibility in terms of *injection* and Cross-Site Scripting (XSS) [297], for instance.

To sum it all up, programmers find it more attractive to provide functional software, caring more about aspect and functionality, than secure software. Additionally, in 2012, developers believed that application development and coding is different in cloud environments. However, that is expected to fade away in 2013 as the only differences are the SOA approaches and the configurations for availability and performance [260]. Furthermore, open-source software is free and has its code exposed, which eases *reverse engineering* and finding bugs to exploit. OpenStack [195], an ubiquitous cloud computing platform for public and private clouds, is a good example of open-source software.

4.2.3 Summary

Table 5 summarizes the security issues discussed in this subsection, which gives emphasis to the software category of the taxonomy. The analysis of the table shows a set of vanguard issues on user frontend and platforms. Thus, cloud environments are, by design, exposed to issues not specific to the technology, but to the business model itself. However, issues related with the software spread to VMM management interfaces, which are an inherent component of the technology.

4.3 Storage and computing security issues

The problem of outsourcing storage and computing responsibilities to a third-party is that customers do not know what happens within the cloud. Because customers do not have their data locally, a plethora of barriers arise. Wang et al. [292] said that storage security has always been an important aspect of the Quality of Service (QoS). Hence, proper techniques and mechanisms are required to efficiently and reliably check data status in two scenarios: before and after being computed, and while being persistently stored. However, Ateniese et al. [19] acknowledged that the main issue of such checking is to verify how frequently, efficiently and securely a storage server, or a group of servers, is faithfully storing customers outsourced data, which is always under the threat of being tampered with by insiders or outsiders [257]. The discussion included below tackles security issues related with *data storage*, *unreliable computing*, *availability*, *cryptography*, *sanitization*, and *malware*.

4.3.1 Data storage

Data storage services, like Dropbox and Google Drive, opt to offer persistent hard storage plans for data. As it is discussed

Table 5 Summary of the security issues and respective studies regarding the software category of the taxonomy

Category	Topic	Issues	Studies
Software	Platforms and frameworks	Isolation, resource accounting, safe thread termination	[232]
		Insecure system calls and deficient memory isolation	[175]
		Bad SDLC approaches	[161]
	User frontend	Internet exposure of frontend interfaces	[3,100,211,281]
		Deficient configurations, unauthorized access	[100]
		Application loopholes, masked code injection	[261]
		VMM management consoles vulnerabilities	[297]
		Programmers beliefs	[260]
		Open-source software, reverse engineering	–

in [180], there is a cloud war going on between cloud providers. Prices are flattening due to the wide solutions available across several providers—there is a competitive landscape out there. In the midst, some even offer bold solutions, such as free space on the cloud without nothing in return. Nevertheless, data is sent, viewed or edited remotely. These three fundamental actions drive where such storage providers are heading. A realm of online collaboration is required to achieve that objective. In fact, Box [36] is a step forth to achieve that objective. However, such a model implies for document owners to delegate, to some extent, authorization permissions to other tenants, creating an even more dynamic environment.

However, the *loss of control* [300] issue yielded by clouds makes it harder to check for data integrity and confidentiality in such an environment. Customers are physically separated from their data and consequently the cloud storage or computing servers, which customer shave no control over them whatsoever. Moreover, the data is somewhere within the server pool, at an unknown location. Because the virtualization layer abstracts resources above, this prevents pinpointing the exact physical location (e.g., storage partition, network port, and switches involved [248]) of the data at a certain moment in time. As a consequence, this unique issue makes it even harder to contain an incident, because isolating or tracking a compromised source implies finding it at forehand.

As discussed in Sect. 3.4, data centers are highly available by ensuring electrical source redundancy and efficient cooling. On top of that, clouds are elastic, meaning that resources are allocated and reused as fit proper. A third step in availability is data redundancy. This means that data is backed up to some other server, which is usually in another data center of the cloud provider. In case of a complete failure of one of the data centers, the data on other data center is still available. However, big players like Google and Amazon have data centers spread over different countries around the world. This is a *multi-location* [310] feature that can bring compliance and legal problems, as data travels across borders (this is further discussed in Sect. 4.9.3).

Subashini and Kavitha [261] pointed out that data integrity is preserved in a standalone database system where Atomicity, Consistency, Isolation and Durability (ACID) properties are ensured. However, clouds are distributed systems with a higher complexity and dynamics, and transactions between data sources must be handled correctly in a fail-safe manner. Auditing is an adequate solution for checking the data state. But, it would not be fair to let one of the entities engaged in the storage agreement to conduct the auditing tasks, because neither of them could be assured to provide unbiased and honest auditing results [291]. Additionally, customers may not have the time, willingness, resources, or feasibility to carry those duties. In such case, they may delegate such responsibility to an optional trusted third-party auditor.

4.3.2 Unreliable computing

Helland [109] stated that many service applications fit within a pattern of behavior. Such service applications have the goal of implementing the frontend for SaaS applications, which arrive via Web service or Hyper-Text Markup Language (HTML) requests. That pattern is composed of a sessions state manager, other services that may be called upon, and cached reference data. As explained in the work, a service call tree is obtained when an application calls another service which, in turn, requests another service, and so on and so forth. Therein, to meet a system-wide SLA, services down the tree are under enormous pressure to meet tight SLAs. Traditional SaaS applications have 300 ms response time for 99.9 % of the total number of requests with a rate of 500 requests per second. A top-down approach reveals ever-tighter SLAs constraints in the call stack, to which the bottom level is the most stringent. Therefore, any delay in one service node can have a snowball effect to services below. Such delay can be perpetrated by *malicious agents*, *downtimes* or *slowdowns* [302], which can result in *dishonest computing* [300]. Moreover, data can be accidentally lost through *administrator errors* in backups, restores or even migrations. For instance, MapReduce, a computing framework for processing large datasets in distributing systems, may output dishonest, inaccurate computational results because of misconfigured or malicious servers. Finding out which machines are compromised is nonetheless a difficult task. Moreover, MapReduce does not have an integrated security model because it was designed to run in a single data center [234].

4.3.3 Availability

Cloud services need to be up and running around the clock to meet the high availability goal. IaaS physical and virtual resources, like databases and processing servers, need to be available in order to support data fetch operations and execute computational tasks of programs, respectively. To this end, architectural changes are made at the application and infrastructural levels to add high availability and scalability. Subashini and Kavitha [261] said that a multi-tier architecture needs to be adopted, supported by a load-balanced farm of application instances, running on many servers. This approach enables DoS attacks resiliency by building software and hardware failure measures in all tiers. Notwithstanding, it is easy for a malicious actor just to rent several services from the same cloud provider and manage them at will. Then, it is possible to have servers processing highly demanding intensive tasks so as to occupy available resources, includ-

ing memory and processing power and time. Although SLAs are agreed to depict the quantity and speed of memory and CPUs, nothing is deterrent to have them occupied at all times in a bogus manner, with fake tasks for instance. At a certain point, resources might be denied to other customers. Nevertheless, such issue is partially allayed by the elasticity feature of cloud environments.

Another issue in terms of availability is related with hardware availability [3]. A single minor glitch can lead to partial of complete blackouts of the systems. So far, ten *cloud outages* of major cloud providers have been reported in various studies [3,17,224]. Those cloud *outages* ranged from several minutes to several hours—paralyzing businesses in general—and happened mostly in 2008 and 2009 on Amazon S3, GAE, Gmail and Microsoft Azure. Nevertheless, in 2011, Amazon EC2 faced an *outage* that affected Netflix and Reddit. The culprits include single bit errors, services overload, programming bugs, protocol blowups, and network glitches. Thus, *outage* events should be negotiated upfront in SLAs to discriminate disaster recovery and backup plans.

4.3.4 Cryptography

Cryptographic mechanisms are many times the most straightforward security measures applied. Nevertheless, they require careful implementation because cryptography does not guarantee complete security. Cryptographic mechanisms rely on the assumption that it is computationally unfeasible to calculate some values, given the result of an operation. Examples are the prime factorization of large numbers and the intractability of the discrete logarithm, both providing the security for the Rivest, Shamir, Adleman (RSA) standard. However, faulty implementations or bad password choices make malicious actors resort to *brute-force* attacks first—a technique that goes through the universe of all possible combinations for a given cryptosystem. The MEGA [168] service encrypts every file at the user end before being uploaded to the cloud. Files are encrypted and checked for integrity by chunks using Advanced Encryption Standard (AES) and Message Authentication Codes (MACs), respectively. A symmetric key of 128 bits is used for these operations. Grobauer et al. [100] mentioned *insecure or obsolete cryptography* and *poor key management* as potential issues. Yu et al. [306] added *faulty algorithms*. Hence, programmers should have these cryptographic concerns in mind when developing SaaS applications and mechanisms for securely storing data and computing programs.

Nowadays, *brute-force* attacks represent a growing threat [257], mostly because they are easier to carry out. Two preponderant factors contribute to this issue: evolving technology and password cracking methods [95]. Nowadays, computers pack greater processing power distributed across various platforms, including multi-core CPUs and Graphics Processing Units (GPUs) with high clock rates. This enables to quickly search—in terms of time complexity—several huge combinatory keyspaces of lower- and uppercase letters, digits and symbols. For instance, it was recently shown that a custom-built 25 AMD Radeon GPU-based cluster with the OpenCL framework can tore through 348 billion password hashes per second [280]. Windows XP passwords can be cracked from just a few minutes up to a few hours, depending on whether Local Area Network Manager (LM) or NT LM (NTLM) security is used. In addition to capable hardware, crackers also rely on advanced techniques that were tuned up over the time, allowing an efficient search of the keyspace universe in terms of algorithm complexity. Massive database password breaches (containing millions of plaintext, hashed, or encrypted passwords) throughout the years have given a structured perspective on user habits when it comes to password choosing and provided the elements to assemble big rainbow tables and dictionary lists in the order of hundreds of millions [1,277]. For example, it is common to see passwords with first capital letters or a name followed by a year (e.g., JohnDoe2012), or to exchange particular letters for similar numbers (e.g., "cracker" would become "cr4ck3r"). The recently hacked LivingSocial company exposed salted and hashed passwords of fifty million customers due to a cyberattack [155]. A vastness of cracking applications is publicly available, including `oclHashcat`, `Extreme GPU Bruteforcer`, `John the Ripper`, `Ophcrack`, `GRTCrack`, and `CloudCracker`.

4.3.5 Sanitization

Sanitization is the process of cleaning or removing certain pieces of data from a resource after it becomes available for other parties. For example, deleting data has been a concern in distributed systems for a while now, to which monitoring, marking and tracking mechanisms have been employed for data discovery [174]. Data sanitization is an important task in order to properly dispose of data and physical resources that are sent to the garbage. For instance, Google has destruction policies to physically wreck hard drives. However, *deficient implementation of data destruction policies* at the end of a lifecycle, may result in *data loss* [34] and *data disclosure* [44], because hard disks might be discarded without being completely wiped [17] or might not be wrecked at all because other tenants might still be using them [100,211]. Hence, one can say media sanitization is hard or impossible due to resource pooling and elasticity in cloud environments.

Since pooling and elasticity entail that resources allocated to one user will be reallocated to a different user at a later time, it might be possible for subsequent tenants to read data previously written. In fact, the media [42] recently reported a case related with sanitization. Basically, cloud *recycling*, as it was termed, consists in reusing a cloud instance previously

used by another customer. What was strange in the case was that of the instance being exposed to massive amounts of network traffic right after being lit up. It should have been zero. After the new customer investigated, it was found that an Internet Protocol (IP) address was maybe cached and that it belonged to an ad company that perhaps did not realized that IP was still part of their live infrastructure. The instance was nonetheless returned by the new customer. This case describes an innocent oversight that could render all cloud safeguards irrelevant if a bad actor happened to gain access to that instance. Pearson [211] said there is a higher risk to customers when reusing hardware resources than dedicated hardware.

4.3.6 Malware

According to FireEye in their Advanced Threat Report—2H 2012 [84], it is stated that malware events occur once every 3 min at a single organization, in average. Moreover, 50 % of malware downloads additional malicious executables within the first 60 s of infection (usually called droppers), Websense says in the 2013 Threat Report [295]. Droppers can also disable local security, prevent updates and perform an inventory of the victim. Malicious code with an adequate payload can be afterward downloaded from bulletproof repositories and may further communicate with Command and Control (CnC) infrastructures in order to become part of a *botnet*. Chen et al. [47] said that *botnets* in clouds are easier to shutdown than traditional ones. Although malware has been around for long, these indicators show off which kind of threat current companies (including cloud providers) must deal with—and the data is worrisome.

One specific issue related with cloud-based storage providers, such as MediaFire or SugarSync, is inherent with the functionality of syncing data across several devices. If malware finds its way into a folder synchronized with such a cloud, then it can spread across the devices that are also configured with that specific account. Additionally, even if endpoint protection like anti-virus agents are installed, and if the agent matches a signature for the malware, which only has about 30–50 % chances of doing so [295], and if it successfully deletes it from the hard disk, which sometimes is not able to, but if it does, then the cloud can just sync the malware back onto the device. Typically, if the first time succeeded, the agent will detect it the following times and, for the enterprise SOC team, that is good news. Surely an outlier will be visible in the monitoring systems as one node is detected with 500–1,000 or more alerts of the same malware. These type of applications typically create temporary hidden folders to sync data, which is the probable location for the malware to be detected in this case. A noteworthy issue from this discussion is the current signature-based anti-virus effectiveness, which is nowadays very low due to the static nature of the signature databases [151] that have to cope with an increasing growth of dynamic malware (this is further discussed in Sect. 4.4.6).

4.3.7 Summary

In the storage and computing category, a new group of security issues literally comes out from the features of cloud computing, namely the resource pooling and elasticity feature as seen in Table 6. Data can be stored and computed at undetermined locations on shared, third-party managed infrastructures, which subsequently brings more obstacles to the ones already prevalent and known to the security community. Flaws in cryptographic methods or implementations and integrity-checking mechanisms are examples of such problems. Additionally, current malware trends render low detection success for their anti-virus counterparts.

4.4 Virtualization security issues

In the light of cloud computing, virtualization lead the way for the wide adoption in the industry. IaaS providers rely on quick deployment of VMs on top of VMMs in their business. The virtualization layer can be thought off as a primary defense in clouds, but are also a point of entrance for attackers as not all virtualized environments are bug-free [17]. To the cloud providers perceptive, a multi-tenant and virtualized approach seems promising in terms of profit, but increases the *co-location* attack surface. VM-to-VM and VM-to-VMM have arisen, and have improved and been refined over time [28]. Although virtualization security in general has been widely studied in the literature [86], assuring perfect logical and virtual isolation has not yet been achieved. Furthermore, virtualization software has been known to contain bugs that allow virtualized code to break loose, to some extent. The following discussion is focused on security issues related with *managing images*, *monitoring virtual machines*, *virtualized traffic*, *Virtual Machine mobility*, *Virtual Machine-level issues*, and *malware*.

4.4.1 Managing images

The majority of the discussion so far has primarily characterized clouds as dynamic networks. Because they are service-oriented and because of their elasticity, allowing to create, modify, migrate, or copy VMs images—a volatile environment in an ever-changing state. However, those VMs features can bring a few problems discussed next.

VMMs allow VMs to be easily turned on, off, or suspended, saving their current state, including running processes, memory, and data. In subsequent boots, previous states are loaded from the images and applications can be run or rerun as normally. Cryptographic techniques,

Table 6 Summary of the security issues and respective studies regarding the storage and computing category of the taxonomy

Category	Topic	Issues	Studies
Storage and computing	Data storage	Collaborative online cloud storage	–
		Loss of control	[300]
		Pooling, data locality	[248,300]
		Multi-location	[310]
		Integrity checking complexity	[261]
	Unreliable computing	Top-down SLAs call stack tightening	[109]
		Malicious agents, downtimes, slowdowns	[302]
		Dishonest computing, administrative errors in backups, restores or migrations	[300]
		Lack of security in computing models	[234]
	Availability	Bogus resource usage	–
		Cloud outages	[3,17,224]
	Cryptography	Insecure obsolete cryptography, poor key management	[100]
		Faulty cryptographic algorithms	[306]
		Brute-force, dictionary and rainbow tables attacks	[1,95,257,277,280]
	Sanitization	Deficient implementation of data destruction policie	[34]
		Non-wiped hard disk discard	[17]
		Hard disk multi-tenant usage	[100,211]
		Resources recycling	[42]
	Malware	Signature-based anti-viruses effectiveness	[151,295]
		Cloud malware syncing	–

namely encryption or hashing algorithms, can face performance obstacles when dealing with those image files, since they can be large-sized [284]. Image files have to be kept in a repository which, even at an offline state, are vulnerable to *theft* and *malicious code injection* [175]. One possible workaround for VM *theft* is to concatenate several images, because it is harder to copy large-sized files combined than one only. This, however, brings even greater obstacles to the cryptographic techniques. Wei et al. [296] provided a study on security risks for an image repository, from the perspective of the repository administrator, the cloud provider and the cloud user. The administrator risks are *hosting and distributing malicious images*. Security properties of dormant images are not constant and degrade over time, because an unknown vulnerability at the time of publishing images may appear later on. Anecdotal evidence expressed the importance of managing images (e.g., scan for *worms*) in order to converge to a steady state, otherwise *infected* VMs can sporadically disseminate malware, an issue named *transience* by Garfinkel and Rosenblum [86]. This also applies for software licenses, where administrators tend to overlook long-lived inactive images because of high *maintenance costs*, including security patches and updates. Luo et al. [156] discussed VM *sprawl*, which is the case where the number of VMs is continuously growing, while most of them are idle or never resumed from sleep, which may cause wasting resources and complicate VMs management. The cloud provider risks *leaking data* if unwittingly publishes images, because images contain fully configured applications and data. Finally, the cloud user risks running *vulnerable*, *malicious*, *out-of-date* or *unlicensed images* stored at an insecure, wrongly administrated repository. The danger inherent to compromised images lies in bypassing perimeter defenses by running an apparently legitimate VM and places it into the cloud network. This also eases the developing and propagation of *malware*, because VMs encapsulate their software dependencies.

4.4.2 Monitoring virtual machines

VMMs are known not to yet be bug-free and, from time to time, a vulnerability comes along, as surveyed by Perez-Botero et al. [212], who presented breakdowns of vulnerabilities for Xen and KVM. In physical systems, OSes trust underlying hardware to a large degree. Likewise, guests on VMs are required to trust virtual hardware and thus the VMM. VMMs can also be nearly transparent, meaning they are hardly detected, thus making VMM-*based rootkits* possible. These comprise the VMM trust model, which depicts a single *point of failure* or *maliciousness*: the VMM [210]. In turn, the general trust is undermined. Moreover, cloning

VMs means their execution does not follow a linear path through time—they can be reversed (restoring snapshots), forked, and subject to nonlinear operations. This is referred to as *lack of monotonicity*, as Pearce et al. [210], pointed out and can raise issues because it breaks the linear operation of programs running within VMs. For example, information stored in databases, logging and monitoring data, or applications configurations are lost when restoring some snapshot. Pearce et al. further said that keeping such data separate from the snapshoting process itself presents potential risks of data storage. *Isolation*, *inspection* and *interposition* [261] are three key VMMs aspects to work on as well. A VM-to-VMM attack consists in gaining access to the underlying VMM through a legitimately running VM managed by that VMM, an attack named VM *escape* [100]. If successful, the attacker can monitor other VMs, including shared resources and CPU utilization, and shutting down VMs. Well-known VM *escape* attacks include SubVirt [140], BLUEPILL [237] and Direct Kernel Structure Manipulation (DKSM) [24].

Garfinkel and Rosenblum [86] and Vaquero et al. [284] elaborated on the fact that monitoring all VMs massively increases *computational overhead* due to the wide range of OSes that can be deployed in seconds, an issue named VM *diversity*. More work is needed to enhance behavioral and introspection VM techniques [51] while having in mind operational cost. Additionally, as VMMs become more mature, recursive virtualization technologies can be required and new security issues may emerge.

Recently, the vulnerability with index CVE-2013-1920 was assigned to Xen. Although updates were quickly released by the vendor and no exploits were found, if successfully exploited, the memory-corruption vulnerability would allow to *execute arbitrary code* within the context of the affected application. Failed attacks could cause DoS nonetheless [246]. This vulnerability is illustrative of the extent and impact of VMMs vulnerabilities and point out the importance in ensuring security because *zero-day* vulnerabilities are rapidly included in crime packs sold at underground markets. Crime packs are also know as crime kits or exploit packs and include the famous BlackHole, ProPack and Sakura [163]. *Zero-day* vulnerabilities consist on vulnerabilities being possibly exploited in the wild without the knowledge of the security community. HyperVM was once exploited through a *zero-day*, and the attackers were able to destruct many Web sites [171]. These examples illustrate how *zero-day* vulnerabilities can affect the virtualization layer.

4.4.3 Virtualized networking

Real, physical and standard Ethernet or radio networks can already be hard to manage given enough disruptions or anomalies. Therein, a relevant aspect in managing real and virtualized networks concerns the kind of traffic they produce and which security policies are enforced at each VMM. Controlling both types of traffic can be defying, because tried-and-tested network-level security might not work in the virtualized network layer [100]. In fact, Vaquero et al. [284] said that network virtualization in cloud environments leads to reduced security as traditional methods, such as Virtual Local Area Networks (VLANs) and firewalls, prove less effective when moved to virtualized infrastructures. Nevertheless, various security vendors now offer their products in virtual form as well, like the Cisco Virtual Security Gateway for Nexus 1,000 V series switch, which can be deployed as a virtual appliance on VMware or a virtual service blade. Because of the nature of cloud services, Grobauer et al. [100] said that standard controls like IP-based zoning can not be applied in IaaS network infrastructures.

Wang and Ng [293] analyzed the impact of virtualization on network performance of Amazon EC2 instances. Both widespread processor sharing and virtualization were pointed as causes for *unstable network characteristics*, namely abnormal packet delay variations and unstable Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) throughput. Such nature brings *limited administrative access* and *network tailoring* which, in turn, can leave network holes. In such a scenario, attackers might be able to reach sensitive portions of the underlying infrastructure belonging to the provider or to other resources belonging to other customers [17].

VMMs typically offer various basic types of networking to child VMs [210]: bridging virtual NICs to physical adapters (appears to be directly connected to the physical network), Network Address Translation (NAT) routing (sharing the IP address of the host), and internal and isolated networking (private network shared with the host). On public IaaS clouds, it is desirable to treat VMs as if they are standard physical servers, thereby bridging VMs networking seeming as the better solution. VMs on Amazon EC2 are publicly accessible through a unique name that is translated into an IP address. A bridged adapter can send, receive, and listen to traffic on the physical network and can occur with little to no traffic intervention from the host (e.g., firewall rules, MAC address, or NAT modifications). This can be an issue in case of *promiscuous mode* where VMs can see all traffic including that not addressed to them [210]. Such possibility is nonetheless dependent on the security policies established on VMMs. In this regard, Wu et al. [298] also identified *packet sniffing* and *spoofing* as threats in virtualized networking environments. Moreover, vulnerabilities in virtualization software, such as virtual switches, can result in *network-based* VM attacks [175]. Pfaff et al. [213] also pointed out the particular issue of securing the dynamic establishment of *virtualized communication channels*, which is aggravated when

SaaS applications and VMs dwell across various IaaS platforms.

4.4.4 Mobility

Due to the cloud pooling and elasticity features, VMs can be easily copied or moved to other servers. This process is usually called VM *cloning* or *template image cloning* [79,100]. This can be troublesome because several of VMs can be running copies of the same image, essentially relying of the same initial battery of software, or the same initial state. Such a copying process can ease the *propagation of erroneous configurations* [123], or even worse, a template image might retain data from the original owner (e.g., secret keys and cryptographic salt values) which can be *leaked* to a new tenant when the VM image is copied. Moreover, because VMs can have multiple copies throughout the network cloud, if an attacker can take one unnoticed, it might be possible to read its contents while trying to break the administrator password. Similarly, because VMs are often created for short periods to serve specific purposes, there might not be a large enough time window to develop a sufficiently unique entropy pool, as Stamos et al. [259] posits on the 2009 edition of the Black Hat conference. An adversary can try to guess entropy pools of other recently created VMs, at least of Linux-based guest OSes. Kirkland [141] said, in his talk on the 2012 OpenStack Design Summit, that VMs are always instantiated with the same initial seed for Pseudo-Random Number Generators (PRNGs) and are sometimes publicly known, at least on OpenStack instances. The problem can, however, be generalized, because cloud instances are instantiated from that so-called VM image template.

The VM *mobility* [86,282,308] feature provides quick deployment of VMs on-the-fly, but also brings various issues. To address them, several security requirements should be checked while VMs are transferred through the network, and when are deployed. However, Oberheide et al. [188] explored a Man-in-the-Middle (MitM) attack on Xen and VMware VMMs during live VMs migration. Live VM migration implies VMs to be running while being migrated. The attack explores three classes of threats: the control plane, the data plane and the migration module. The tool `Xensploit`, capable of exploiting VMware and Xen, was developed and explained in their work. Furthermore, Zhang et al. [308] outlined a Time of Check to Time of Use (TOCTTOU) vulnerability and *replay* attack.

4.4.5 Virtual machine-level issues

As discussed next, because VMs in IaaS infrastructures are at the mercy of the customers, there are a bonanza of potential severe threats. As Jasti et al. [123] pointed out, VM *hopping* consists in maliciously gaining access to different VMs belonging to other customers by exploring the VM-to-VM or VM-to-VMM attack vectors.

Mostly known as a *cross*-VM attacks [300], the prerequisites for these attacks are to have two VMs running on the same physical host and to know the IP address of the victim. With standard customer capabilities, both requisites can be met, according to Ristenpart et al. [227]. If an attack is successful, it is possible to monitor resource usage, modify configurations and files, or leak sensitive data. Because VMMs are likely to place several VMs co-resident, the probability and danger of VM *hopping* are high, setting the severity of this issue as high. Ristenpart et al. [227] and Bugiel et al. [39] demonstrated in 2009 and 2011, respectively, the existence of *cross*-VM *side-channel* and *covert-channel* vulnerabilities in Amazon EC2. *Side-channel* techniques passively observe data flowing without interfering, whereas *covert-channel* methods actively inject bits to acquire some sort of information [47]. Zhang et al. [309] were able to extract a 4,096 bit ElGamal public key off a co-resident VM handled by a Xen VMM, from which a partial private key was able to be computed. The remainder of the private key could be obtained through a *brute-force* attack. The *side-channel* attack exploited square-and-multiply algorithm instructions stored on a L1 instruction cache. Besides needing a co-residing VM, the attack also required a machine learning algorithm to be trained on the target hardware and the victim to be decrypting an ElGamal ciphertext using `libgcrypt` v.1.5.0. Okamura and Oyama [190] provided a *covert-channel* attack by using CPU load, which was able to encode information. Xu et al. [301] have exploited the L2 cache *covert-channel* to leak small useful information, such as private keys. Aviram et al. [20] regarded *timing side-channels* as insidious security challenges because they are hard to control, provide the means to steal data, can only be detected by the cloud provider, and can undermine efficiency. Moreover, Rocha and Correia [231] demonstrated a series of simple-to-execute *malicious insider* attacks on VMs. Plaintext passwords and private keys were able to be exfiltrated from VM memory dumps and memory snapshots, respectively, while arbitrary commands were possible to be executed in a VM backup copy by following a sequence of steps in Domain0. Moreover, data was possible to tamper with by exploiting VM relocation. A series of studies [265,266] further show the VM-level security state, but this time by exploring the memory deduplication mechanism. This mechanisms reduces physical memory usage in shared environments, therefore being appropriate to virtualized contexts. The mechanism was exploited in the form of *memory disclosure*, allowing one to detect applications or files on co-residing VMs. In addition, Jensen et al. [130] looked into the *cloud malware injection* attack that consists in injecting malicious services or VMs into clouds, serving any particular purpose of the attacker. It is initiated by injecting a service

into the SaaS or PaaS models, or a VM into the IaaS model. Secondly, the cloud system must be somehow tricked to execute the service or VM. Ultimately, authentic user requests are redirected to the malicious instance and the code written by the attacker is executed, which can compromise the overall security state.

One particular issue is due to the abstraction layer that virtualization creates between VM and underlying hardware. This is especially important for cryptographic purposes, specifically for entropy gathering daemons that rely on hardware interrupts to generate entropy pools with strong bits. In turn, Random Number Generators (RNGs) use such entropy pools to potentially generate cryptographically strong random numbers that provide robustness to cryptographic material like Secure Shell (SSH) keys and Domain Name System Security Extensions (DNSSEC). Problems related with low entropy on Amazon EC2 instances have been reported [12], pointing out the hidden vulnerability in Xen platforms. VMware and VirtualBox have also fallen to the pit [207,289], and others on undisclosed VMMs have been reported [40]. Attacks have also been demonstrated [136]. Because hardware interrupts cannot be supplied, not only the strength of the entropy is potentially affected but also the generation speed, leading to the depletion of entropy pools. In theory, guest OSes only have access to network interrupts [136,259,273].

Ristenpart and Yilek [228,304] enlightened on the VM *reset* vulnerability. When a VM snapshot is reused, Transport Layer Security (TLS) sessions were able to be compromised and secret Data Signature Algorithm (DSA) authentication keys were extracted in the authors experiments. The exploits were shown on VMware and VirtualBox and were possible due to randomness repetition. In other words, the state of the entropy pools of the OSes was rewinded, creating a setup to henceforth predict future RNGes states, such as the `/dev/random` or `/dev/urandom` devices of Linux OSes.

The attacks described herein, mainly the *side-channel* attacks, are not easy to perform and are not for the average skilled person. As pointed out by Green [98], the *side-channel* threat has long been discussed by cloud security experts, but has largely been dismissed by providers. The reason is simple, turning theory into practice in this area seems surprisingly difficult. An exceptionally set of skills and knowledge are required to carry them out to completion. Noisy information produced by other VMs and the VMM itself or the fact that VMs Virtual Central Processing Units (VCPUs) are systematically bounced from one CPU core to another may foil the attacker [98]. After all, it is the IaaS providers who own the virtualization infrastructures, therefore having the measures to limit what at least a *malicious insider* can do [230].

4.4.6 Malware

Even thought virtualization opened the door for cloud computing to thrive, it has also transformed how security experts like forensicators operate on a daily basis. Because VMs are supposed to be well isolated, and because it is possible to easily take snapshots, rollback or even delete them, they are suitable for malware analysis. The rollback feature can be problematic because it can out-date anti-viruses or firewalls installed on guests [26], or even return entropy pools to past states, which, if known to an attacker, can henceforth predict pseudo-random numbers. VMs provide environments that can be subjected to disruptions caused by malware in a fail-safe manner. Virtualization is also used in combination with sandboxing techniques. Besides allowing separating running programs, sandboxing has been particularly used for automated malware analysis systems, such as the popular Cuckoo system [65]. Other solutions are available, like the Browser Sandbox [258], which launches Web browsers in sandboxed virtual environments.

Because IT is moving to the cloud, it is expected for malware to follow. Despite the advantages virtualization and sandboxing provides for malware analysis, their evasion techniques have changed [256]. For instance, the popular Conficker malware has, since version .B, included the Store Local Descriptor Table (SLDT) instruction [143], which is used for VM detection. According to Ortega [198], new evasion techniques can be grouped into VM-aware, sandbox-aware and debugger-aware. This means that malware tries to detect whether it runs under a virtual environment, a sandbox environment, or under debug surveillance, respectively. Automated analysis is *dormant* and it is possibly executed in separate isolated servers (e.g., a malware laboratory), hence being devoid of human interaction—keyboard strokes and mouse movement or clicks. Malware exploits this setup by looking for inactivity signs. The UpClicker *trojan* was analyzed by FireEye malware analysts [251], and it was found out that the *trojan* hooked the mouse. If activity would be detected, the malicious code would be normally executed, otherwise it would remain silent—and therefore immune to analysis. Furthermore, a more cautious type of malware can postpone Internet communications by minutes, hours or weeks deliberately to bypass short-term sandboxing analytics [295]. Other techniques include checking for registry values, checking for video or mouse drivers, or even executing especial assembler code [250]. Nonetheless, in contrast with the previous discussion, the more bold malware Trojan.Maljava, as detected by Symantec products, copies itself onto VMware VM image files after mounting them with VMware Player [134]. It is believed to be the first malware to spread onto VMs, hence being a leap forth for next generation malware.

Older malware evasion techniques include polymorphism and oligomorphism [152,173,183,191]. These consist in using irreversible operations, such as *XOR* and encryption, respectively, to obfuscate and transform malicious payload. They are, nonetheless, statically detectable with high probability by means of statistical or semantic mechanisms. A more dangerous morphing technique has been baptized as Frankenstein [173], a malware with a metamorphic engine that stitches together binaries from other benign-looking software that the malware scans in memory. In short, a mutant malware composed of components from honest program parts. The malware trends outlined above point out that malware development is adapting and evolving to virtualized environments, on which malware writers are putting significant effort on evasion [84]. Chen et al. [46] have thoroughly characterized the prevalence of malware evasion methods by executing 6,900 malware samples under different environments. Tests showed 40 % of malware reducing malicious behavior under VM or debug environments. In the same study, a technique using TCP `SYN` messages for remote fingerprinting of VM environments is presented. Such technique is useful to malware for avoiding monitoring systems like honeypots and prolong their prevalence. The technique is further able to distinguish between VMMs types, namely VMware and Xen.

4.4.7 Availability

Similarly to the availability issues discussed in Sect. 4.3.3, a DoS can be tempted against the VMM layer. One or more legitimate VMs can be used to occupy as much as possible of available resources. In addition, one can try to instantiate as many VMs on the same VMM in order to impede the VMM of handling more VMs locally [282]. At this point, support to other VM instances would be denied. A threshold for resource and VM allocation per customer should nonetheless be defined along with proper configurations, therefore mitigating this issue.

4.4.8 Summary

The virtualization category depicts the major brunt in cloud environments security issues, the wider gap and, at the same time, the tallest barrier to overcome in order to achieve a securer cloud. Table 7 contains the summary of those security issues. The spectrum of security issues starts in the very isolation property of VMMs and extends to the pooling and elasticity features, yielding issues of dormant images to VM diversification and distribution, of VM segregation to VMM vulnerabilities to VM hopping, and of newly virtualized network traffic to a mobile VM-enabled cloud. Clearly, VM-level issues dominate; most notably the cross-VM attacks, which points out the yet insecure nature of virtualized OSes and inherent virtualization technologies. Moreover, malware techniques are beginning to shift from static approaches to dynamic, VM-aware, methods.

4.5 Internet and services security issues

Cloud infrastructures are not only composed by the hardware where the data is stored and processed, but also by the path to where it gets transmitted. In a typical cloud scenario, data is transmitted in a large number of packets from source to destination through umpteen number of third-party infrastructure devices and links [227,306]. Because the Internet is normally used as the transmission medium, one has to assume its unsafety and inherent problems. Since the appearance of Web 2.0, a new class of threats emerged along with the people learning how to exploit them. Thus, cloud environments inherit many known issues from the Internet, such as MitM attacks, IP *spoofing*, *port scanning*, *packet sniffing*, *malware*, and *social engineering*. So, even if a significant number of security measures are placed within the cloud, the data is still transmitted using Internet and standard Wide Area Network (WAN) technologies. Moreover, cloud access technologies can vary from service enabled fat clients to Web browser-based thin clients [130], being the latest the most commonly used nowadays [108]. In fact, cloud Web services are required to be used and managed over the Web and a browser is most suitable application to deliver this management interface to the end user. The following discussion includes security issues related with *Advanced Persistent Threats and malicious outsiders*, *protocols and standards*, *Web services*, *Web technologies*, and *availability*.

4.5.1 Advanced persistent threats and malicious outsiders

The security industry has embraced the term Advanced Persistent Threat (APT) [37,149,256] to refer to attacks of a higher degree of sophistication, hence the advanced. In addition, such attacks are more targeted, per se implying a pre-determination of the targets and, most importantly, an objective for attacking. Persistence is a characteristic of these attacks, meaning that attackers just do not run off when they find difficulties in bypassing systems. An APT is strongly related with an attack model consisting of three phases [256]. The first phase is the intelligence gathering phase, on which an attacker passively, semi-passively or actively searches for intelligence. In the passive mode, some public or private intelligence sources can be searched. In the security community, searching public sources is known as Open-Source Intelligence (OSINT) gathering. The Réseaux IP Européens Network Coordination Centre (RIPE NCC) is one of the five Regional Internet Registries (RIRs) that provide Internet resource allocations, registration services and coordination activities that

Table 7 Summary of the security issues and respective studies regarding the virtualization category of the taxonomy

Category	Topic	Issues	Studies
Virtualization	Managing images	Large-sized images cryptographic overhead	[284]
		Image theft and code injection	[175]
		Dormant and overlooked image repository	[296]
		VM transience	[86]
		VM sprawl	[156]
	Monitoring virtual machines	VMM single point of failure or maliciousness, untrusted VMM components, transparent VMM-based rootkits lack of monotonicity	[210]
		VMM isolation, inspection, and interposition	[261]
		VM escape	[100]
		VM diversity, VM monitoring overhead	[86,284]
		VMM zero-day vulnerabilities	[171]
	Virtualized networking	Twofold traffic, limited access and network tailoring, inapplicability of standard security approaches	[100]
		Network security devices effectiveness in virtual networks	[100,284]
		Unstable network characteristics	[293]
		VMs adapters promiscuous mode	[210]
		Packet sniffing and spoofing	[298]
		Virtual devices software vulnerabilities	[175]
		Virtualized communication channels	[213]
	Mobility	VM cloning	[79,100,210]
		VM mobility	[86,282,308]
		Propagation of erroneous configurations	[123]
		Live VM migration MitM attack	[188]
		TOCTTOU vulnerability and replay attack	[308]
	VM-level	VM hopping, cross-VM attacks	[123,300]
		Side-channel attacks	[20,227,309]
		Covert-channel attacks	[39,190]
		VM data exfiltration attacks	[231]
		Memory deduplication exploits	[265,266]
		Malware injection	[130]
		Entropy generation strength	[136,259,273]
		Entropy depletion	[12,40,207,289]
		VM reset vulnerabilities, randomness re-usage	[228,304]
	Malware	VM rollback	[26]
		Malware evasion techniques	[46,143,198,251,250,256,295]
		Malware spreading onto VMs	[134]
		Metamorphic engines	[173]
	Availability	Bogus VM usage, VMMs capacity to handle VMs	[282]

support the operation of the Internet globally. The RIPE NCC has a database [225] of all the IP addresses allocated to some specific Internet Service Provider (ISP). Useful information like the subnet mask, associated Autonomous System (AS), country, ISP name, and address of the headquarters can be extracted by querying a simple IP address. Other sources like social and professional networks can be look into to correlate information across several places, a

practice recently named *doxing* [96]. In the semi-passive mode, an attacker can generate traffic but without raising suspicions. That includes performing Domain Name Service (DNS) or `WHOIS` queries. Many online tools ease the job, like Network-Tools [182]. In the active mode, an attacker can perform more bold *reconnaissance scans* (e.g., *port scan*) to map the target network. The second phase is the threat modeling phase. An attacker maps the target network and assesses which way and techniques are best to adopt. In the last phase, an attacker finally performs the attack and exploits possibly found vulnerabilities.

The most interesting and perhaps the most outspoken APT incident ever recorded was recently made public by Mandiant, a cybersecurity company. The unprecedented report [157] contains all the details about the APT and was based on several years of investigation. It exposed China to have one government-supported cyberespionage unit located in Shangai active in APT operations since at least 2004. The espionage campaign compromised 141 companies spanning 20 major industries across the globe, stealing hundreds of terabytes of data in total. The security community received the report with charisma, raising their alertness to APT signs.

Nowadays, a cyberwarfare state is in place. State-sponsored malicious cyberactivity like the operations exposed by Mandiant has clear goals: *espionage* or *profit*. *Data exfiltration*, like intellectual property or business secrets, can have serious impact in enterprise survival. In fact, the Mandiant report put the cyberworld onto notice. Since then, the Pentagon of the USA has said that will create thirteen teams capable of offensive cyberoperations [38]. The so-called rules of engagement will provide a framework for how to best respond to a plethora of cyberattacks, including attacks on private companies. Hacktivism, on the other hand, is mainly related with a kind of political protest, and common aftermaths include Web site *defacements*, Uniform Resource Locator (URL) *redirection* and DoS. Hacktivists should, nonetheless, not be considered less a threat.

As perceivable from the discussion above, intelligence gathering can be the most important phase. Information on how or where to attack can be critical for the success of an APT attack. Thus, one should pay attention to what kind of information related with the enterprise environment is publicly available. Sood and Enbody [256] mentioned the exploitable state of AWS, which raises further concerns for cloud-enabled enterprise environments. However, Amoroso [13] said that APT effects are diminished in a mobility-enabled secure cloud. If the design goal of a multi-tenant environment is to constrain a small perimeter to only the resources supported, then, in theory, a malicious outsider can gain access to those resources only. However, this article yet discusses several security breaches from small, supposedly isolated perimeters like a VM.

4.5.2 Protocols and standards

Because the TCP/IP model is the basis for communicating in the Internet, the protocols and standards of its stack are important to have in mind, but not only, in the Web-based cloud environments. Dynamic Host Configuration Protocol (DHCP) and IP (e.g., IP *spoofing*) are among known vulnerable protocols, along with DNS (e.g., DNS *cache poisoning* or DNS *spoofing* [203]), which may enable *network-based cross-tenant* attacks [175]. For instance, *botnets* usually abuse the fast flux DNS characteristic to their own benefit. Fast flux DNS features a load balance technique to alternate IP addresses related with a single host name, therein redistributing traffic among various servers. Traditionally, those IP addresses do not change very often. However, to hinder discovering such servers tied to a domain (e.g., proxies or CnC servers), IP addresses are swapped in and out with short Time-to-Live (TTL) values in a round-robin fashion.

The HyperText Transport Protocol (HTTP) is by design a stateless protocol and does not guarantee delivery. To address this, Web applications usually implement session handling techniques, many times being vulnerable to *session riding* or *session hijacking* [100]. This threat is of utmost importance for SaaS applications. Hunt [116] elucidated on the fact that many Web sites wrongly implement HyperText Transport Protocol Secure (HTTPS). The threat lies when HTTPS safeguarded content streams are mixed up with HTTP streams on a main page served over HTTPS or HTTP. Certain Web sites implement HTTPS in sensitive forms, like a login form, and then the rest of the session is maintained over HTTP. This does not guarantee security at all because the session cookie can be sniffed in plaintext. This issue is called *mixed content* and Mozilla Firefox will have it blocked by default in version 23 [272]. Cookies can be used for any purpose, including *cookie poisoning* and *impersonation* attacks [203]. HTTPS can be enforced to be in an always-on state by means of local browser addons. HTTPS Everywhere [81] and ForceHTTPS [122] are good examples. More recently, the HyperText Transport Protocol Strict Transport Security (HSTS) is currently under proposed standard on the Request for Comments (RFC) 6797 [111], which consists of a mechanism to declare Web sites accessible only under secure connections or to instruct user agents only to interact with Web sites under secure connections. HSTS is an enhancement of ForceHTTPS. However, even with HTTPS, cookies can be exposed to various attacks. Several attacks on TLS have been discovered over the years, like the BEAST, CRIME, and Lucky 13 attacks on TLS in Cipher-Block Chaining (CBC) mode, which are now ineffective if certain countermeasures are applied. Heninger et al. [110] performed an Internet-scale study in the pursuit of weak TLS and SSH hosts. They were able to scan 12,828,613 TLS and 10,216,363 SSH devices, from which several alarming

key findings were found. Surprisingly, 0.75 % of TLS certificates shared keys, 0.50 % of TLS hosts and of 0.03 % of SSH hosts RSA private keys, and 1.03 % of SSH hosts DSA private keys were obtained. In either case, the guilty party was bad randomness. More recently, the Rivest Cipher 4 (RC4) stream cipher of TLS was broken by a group of researchers [7] by applying a combination of a statistical procedure with biases found in RC4 keystreams. Furthermore, Marlinspike [159] showed how to perform a MitM in HTTPS-based connections. By exploiting a flaw in checking the `Basic Constraints` field of X509v3 certificates, the author built a tool named `sslsniff` that creates on-the-fly certificates for the domains a user is accessing to and proxies data through. Basically, the `Basic Constraints` field indicates whether or not the certificate belongs to a Certificate Authority (CA). Because this field is many times overseen and not validated (e.g., Web browsers), a forged certificate can be created for any domain without the requirement of passing through a CA. This attack has the prerequisite of having a CA to sign a certificate owned by the attacker for a legitimate domain. Other forged certificates would then be created using that legitimate domain certificate. Marlinspike also built `sststrip` [160], a MitM tool that transparently maps HTTPS links and redirects into HTTP links or homograph-similar HTTPS links, thus exploring the aforementioned *mixed content* issue. Prandini et al. [214] used the tool to provide practical examples.

For the rest of the service delivery models, Simple Object Access Protocol (SOAP), REpresentational State Transfer (REST), and Remote Procedure Calls (RPCs) are used for PaaS Web services and APIs, while remote connections, VPN technology, and File Transfer Protocol (FTP) are used for IaaS services [175]. REST defines an lightweight architectural style of the Web to which HTTP tightly adheres because of its basic HTTP verbs like `GET`, `PUT`, `POST`, or `DELETE` [4]. On the other hand, SOAP offers a more complex service contract, data structures and APIs, which are manifested through Web Service Definition Language (WSDL) files. Because of the simpler operation REST provides, in contrast it is not adequate for very complex systems. Nonetheless, REST is thought as the successor and replacement for SOAP-based Web services [30].

4.5.3 Web services

Subashini and Kavitha [261] stated that due to the clouds SOA approach, the problem of data integrity gets magnified when compared to former distributed systems. Web services normally expose their functionality via eXtensible Markup Language (XML) and APIs. HTTP fails to guarantee data integrity, and most SaaS vendors deliver their Web services APIs without transaction support, which further complicates the management of data integrity across multiple SaaS applications. Moreover, WSDL is a language standard for describing the functionality of a Web service, specifying how it can be called, what parameters are expected for input and the return values. Related with this is the *metadata spoofing* attack, which consists in reengineering metadata descriptions of, for example, WSDL documents by first establishing a MitM [125]. A forged WSDL can permit invoking other operations, not specified in the original document to, for instance, create user logins [130]. If an administrator account is created, the attack can have greater impact. *Metadata spoofing* attacks are easily detected if sound methods are used. Notwithstanding, WSDL documents in cloud environments are often dynamically accessed, drastically raising the potential spread of forged files and, consequently, the probability of successful attacks.

Researchers have paid attention to Web services security even before clouds emerged. McIntosh and Austel [166] studied XML Signature element wrapping attacks, shortly called *wrapping* attacks [130] or *rewriting* attacks [218], and respective countermeasures. *Wrapping* attacks consist in rewriting SOAP messages, captured via *eavesdropping*, by injecting wrapper and forged XML fields to access target resources. The SOAP envelopes maintain valid signatures for the original documents requested by the user, thus allowing the services to execute modified requests. Gruschka and Iacono [102] discovered that Amazon EC2 was vulnerable to a variation of *wrapping* attacks in 2009. After capturing legitimate SOAP user messages, correctly signed, it was possible to perform an arbitrary number of EC2 operations. On the same track, Jensen et al. [125] described *SOAPAction spoofing* as a Web services attack to modify HTTP headers in order to invoke operations different from the ones legitimately specified. Successful examples of both *SOAPAction spoofing* and XML *injection* attacks are presented on a .NET Web service. Another attack entitled WSDL *scanning* is addressed in various studies, such as [77,125]. It consists in discovering and fingerprinting Web services, ultimately to find omitted, confidential operations, supposedly available only to administrators. It should be noticed that the previously described attacks require a mechanism to somehow capture messages in transit by establishing a MitM for eavesdropping purposes. These attacks can also have a variety of aftermaths, including *data leakage* and *access to unauthorized resources*.

4.5.4 Web technologies

Cloud frontend services are accessed via Web-based user agents, and the conventional Web browser is yet the preferred choice. Most intelligence reports show that Web sites hosting malware have been growing systematically. Malicious Web links grew by almost 600 % [295]. Because of the increase in the number of connected people and devices to the Web,

evil attackers have focused on this attack vector. The plot of the most common Web vulnerabilities in the Open-Source Vulnerability Database (OSVDB) Web site [199] over the years demonstrates that XSS has topped all others for some time now. HP in its 2012 Cyber Risk Report [113] also shows that XSS ranked significant positions in the findings of the report.

The non-profit Open Web Application Security Project (OWASP), an organization dedicated to the widespread of good application security practices, has been putting effort to provide guidelines for building secure Web applications. The Top Ten Most Critical Web Application Security Risks [200] of 2010 showed that *code injection* tops the ranking, whereas XSS stands in second. Although the OWASP top for the year 2013 is still a release candidate [201], *injection* maintains the first position while XSS is surpassed by *broken authentication and session management*. The latest includes the *cookie theft* issues discussed in Sect. 4.5.2. *Injection* weaknesses are perhaps most known by the form of Structured Query Language Injection (SQLi). A variant of SQLi is known as *blind* SQLi, which consists in injecting Structured Query Language (SQL) code without feedback from the systems. Panah et al. [203] explained the *hidden field manipulation* attack, which consists in altering HTML hidden fields to whatever the attacker desires. HTML hidden fields are typically used for exchanging data from a form page like a login form, hence a browser, to a Web server, and programmers usually use them for control data.

Malicious Web sites typically appear to be completely legitimate and show no outward indicators of their hidden malicious nature. Such Web sites can be compromised by exploring vulnerabilities in the applications. Thus, because of a faulty SaaS application, the underlying physical host can become compromised and, eventually, infect other hosts or provide a way into cloud environments. Nevertheless, a great part of Web sites is compromised for *phishing* purposes. *Phishing* sites heavily target the financial industry, according the Microsoft Security Intelligence Report Volume 14, which agglomerated data from July through December of 2012 [170]. Another threat related with compromised Web sites is a more insidious one. The *watering hole* attack, as it is termed, consists in patiently waiting for prey to fall onto a previously compromised Web site and then infect the visiting victim with *drive-by* malware. The infection is carried on by exploiting a *zero-day* vulnerability on the device of the victim. Because of a *zero-day user-after-free* vulnerability, at the time, Microsoft Internet Explorer versions 6, 7, and 8 were being exploited in the wild with a *watering hole* attack [268]. Because it was a *zero-day* vulnerability, the stealthiness and undetectability of the attack was high, pointing the importance in choosing wisely the technologies to build SaaS applications and the browsers to access them.

It is a growing trend to see employees browsing social networks, personal email accounts, and other online applications during work time. Therefore, is it more probable for malware to penetrate into the enterprise perimeter through the Web. Thus, it is a risk for companies when employees browse the Internet and in the meanwhile access backend services or cloud applications. In this scenario, the malware can capture login credentials or other sensitive information. Malware installs itself in devices by exploiting plugin vulnerabilities, but mostly browser vulnerabilities [267]. In terms of plugins, the widely deployed Adobe Flash Player and Acrobat Reader are among the top, along with Oracle Sun Java. In fact, Oracle has recently released a critical patch for Java [196], which addressed 42 distinct vulnerabilities that were frequently discovered in short periods of time. In terms of Web browsers, Apple Safari, Google Chrome and Mozilla Firefox constitute the top three for 2012. A series of sophisticated attacks known as Man-in-the-Browser (MitB)MitB attacks [33,66,221] explore aforementioned issues related with plugins or the browser itself, placing taps between the browser security layer and the user. MitB attacks can have any specific purpose. URLzone, Torpig and Zeus are malware examples for MitB attacks.

4.5.5 Availability

Whether it is hacktivism or an act of cyberwarfare, DoS attacks are commonly seen nowadays. Provoking such a state to a single company can paralyze its daily business and, consequently, lose money. For instance, Blue Security folded its anti-spam service called Blue Frog after being under mass mailing spam [146]. Data centers have a massive number of resources in an elastic connected pool. Hence, proper link bandwidth is required to support great amounts of network traffic. However, Cisco [53] identified *bandwidth under-provisioning* as one of the main data center issues. Large server cluster designs are commonly under-provisioned with factors of 2.5:1 up to 8:1, meaning that the network capacity of data centers is less than the aggregate capacity of the hosts inside the same subnet. So, because clouds can house data of many different businesses and because data centers are prone do *bandwidth under-provisioning*, *flooding* attacks can have an even greater impact than on a single enterprise. Nonetheless, induced-DoS states can be achieved through various attack vectors.

In the case of *flooding* attacks, the impact is normally dependent on available bandwidth, processing power and memory. For example, a *botnet* can be used to send, in a successive and quick manner, millions of TCP `SYN` messages to the target server, therein creating a Distributed Denial of Service (DDoS) attack. Three scenarios are possible. In the first scenario, the server overloads by either process-

ing a single malicious request that expects to exploit a vulnerability or processing a massive number of requests. In the second scenario, a network link is fully saturated with bogus requests belonging to the attack and thus reaches its bandwidth capacity, ceasing honest connections. In the third and last scenario, one or more intermediate routers also become saturated to a large amount of bit rate processing per second—no matter what kind of intelligent software is installed, 11 Gbps is always greater than a 10 Gbps router port.

In the cloud computing context, DoS attacks can be grouped into direct and indirect. Direct DoS attacks imply a predetermination of the target service host machine. Possible collateral damage consists in denying other services being hosted on the same machine or network, therein creating indirect DoS. A worst case scenario is known as *race in power* [126]. Cloud systems may react to machines overwhelmed by *floods* by relocating services to other machines, thus propagating the workload—and the attack—to other servers [129]. At some point, aiding systems can host the *flooding*, putting both systems off against each other, both aiding one another with resources until one finally gives in and reaches a full loss of availability state.

Liu [154] described a new form of cloud DoS. The goal is to starve an uplink bottleneck found in the topology with minimal cost. Gaining topology information is important to maximize the attack effectiveness and identify exploitable links. It requires to gain access to enough hosts within the target subnet to produce, preferably, UDP traffic upwardly through the uplink. By using UDP traffic, the attack has the side effect of starving other TCP sessions that back off due to congestion handling mechanisms.

Another form of DoS attacks is known as *resource exhaustion*. Jensen et al. [125] provided a list of *resource exhaustion* attacks on Web services. The *oversize payload* attack has the objective of increasing memory usage of XML processing when parsing XML objects into memory Document Object Model (DOM) objects. The authors observed an increase in memory consumption with a factor of 2:30 for common Web service frameworks. An example of an attack on Axis Web services was presented, which resulted in an out-of-memory exception. Another attack named *coercive parsing* exploits namespace vulnerabilities in XML parsing with the purpose of overusing the CPU. Yet again, an example of an attack on Axis2 Web services was presented, which caused CPU usage of 100 %. Moreover, the *obfuscation* attack aims at overloading the CPU and increasing memory usage. With the same objectives, the *oversized cryptography* attack exploits buffer vulnerabilities and encrypted key chains. Other issues are mentioned in the study, such as the *WS-Addressing spoofing* attack [127] in the Business Process Execution Language (BPEL) [125], and *flooding* by using XML messages [48]. The same authors argued that distributed XML-based DoS attacks may pose serious issues to clouds in the future.

Distributed Denial of Service (DDoS), an old flavor in the security community, have yet again become increasingly popular. The most fierce DDoS attack in the history of the Internet claims to have caused congestion worldwide. Spamhaus, a spam tracker company that provides Realtime Blacklists (RBLs), after posting [124] about being under attack, it was found out that CloudFlare, a Web performance and security company, aided in the attack mitigation [216]. The media went frenzy days later when CloudFlare posted more details of the incident [215], claiming that the attack peaked 300 Gbps according to a tier 1 ISP. The attackers resorted to a DNS *reflection* and *amplification* attack. The attack is initiated by sending spoofed DNS `ANY` queries with approximately 64 bytes in size to thousands of open-revolvers spread throughout the Internet. At this point, those servers send responses to the spoofed IP address with over 3,000 bytes in size, culminating in an amplification ratio of over 50 times, approximately. The attack fluctuated between 30 and 120 Gbps at the beginning. After CloudFlare diluted the bombardment throughout its 23 data centers around the world by using anycast, the attackers changed strategy and targeted the tier 2 and tier 1 ISPs that delivered bandwidth to CloudFlare, on which the 300 Gbps peek was registered. The source of the attack was most likely a *botnet* or a cluster of servers [215]. This case illustrates the *flooding* trends the Internet is now witnessing. Low bit rate *flooding* attacks are diminishing, while high bit rate attacks are steadily rising [14].

Anecdotally, the Distributed Denial of Service-as-a- Service (DDoSaaS) [178] term has emerged in recent months and it partially characterizes what cybercriminals are nowadays offering, in particular DDoSers. It is case to say, *what is old is new again* [56], but with another rate magnitude and techniques. Prolexic, an anti-DoS company, reported an increase of 718 % on average bandwidth attacks on its clients, moving from 5.9 Gbps in Q4 2012 to 48.25 Gbps in Q1 2013 [217]. More concerning is the average 32.4 millions of packets per second finding. Such a high packet rate level can impact both mitigation gear and routers. The use of null routing, or blackholing, also increased to counterattack such issues, but it is not a viable or acceptable long-term strategy. There is one other way of exploiting the cloud business model and have VMs participate in a DDoS. Nowadays, quick, cheap and easily available VPSes can be used for malicious activities [56]. That includes buying several VPSes on bulletproof hosting providers and then use them as bots for DDoS or for sending massive amounts of email spam. Bulletproof hosting providers allows their customers leniency in the contents and usage of their purchases. More worrisome, it is common to see

Table 8 Summary of the security issues and respective studies regarding the Internet and services category of the taxonomy

Category	Topic	Issues	Studies
Internet and services	APTs and malicious outsiders	Intelligence gathering, publicly available information, reconnaissance scans	[256]
		Doxing	[96]
		State-sponsored malicious cyber activity, espionage, data exfiltration	[157]
		Hacktivism	–
	Protocols and standards	Vulnerable communication protocols, network-based cross-tenant attacks	[175,203]
		Session riding or session hijacking	[100,116]
		Mixed HTTP and HTTPS data streams	[116,214]
		Bad randomness usage in cryptographic keys	[110]
		Cookie theft, cookie poisoning, impersonation attacks	[203]
		TLS attacks, cookie theft	[7,159,160]
	Web services	HTTP statelessness, APIs transaction support for integrity	[261]
		Metadata spoofing attacks, WSDL documents dynamics	[125,130]
		XML SOAP wrapping attacks	[102,125,166,218]
		WSDL scanning	[77,125]
	Web technologies	Infected Web sites growth	[295,170]
		XSS vulnerabilities	[113,199–201]
		Injection, broken authentication and session management	[200,201]
		HTML hidden field manipulation attack	[203]
		Watering hole attacks, drive-by malware downloads, plugin and browser vulnerabilities	[267,268]
		MitB attacks	[33,66,221]
	Availability	Bandwidth under-provisioning, VPSes bots	[53,56]
		Direct and indirect DoS, race in power	[126,129]
		UDP uplink flood attack	[154]
		Resource exhaustion attacks	[125,127]
		XML flooding attacks	[48]
		DNS reflection and amplification attack	[216]
		Mobile API consumption	[192]

bulletproof hosting providers selling unlimited bandwidth usage.

Finally, O'Neill [192] enlightened on the problem related with API consumption by mobile applications. In contrast with a browser, mobile applications consume services via cloud APIs. If an attack effectively achieves a DoS state on those APIs, customers become unaware of it. The perception of the applications running on smartphones or tablets is different to the end user because the applications themselves still run, whereas a Web page would show up an error, such as HTTP `404`. End users may simply blame network congestion problems or mobile coverage, without ever suspecting of offline APIs.

4.5.6 Summary

The issues of the Internet and services category are summarized in Table 8. Because most clouds demand to be publicly accessed from any location, the fourth category of the taxonomy puts the focus on the Internet and its most prominent threats to enterprise computing, paying particular attention to the magnified endangerment of cloud environments, when exposed to such threats. The service-based utility computing relies on Web technology, whose issues have long been known to the community. From flawed communication protocols not adequate for supporting clouds to vulnerable Web technology, ranging from basic HTTP statelessness to com-

plex XML-based attacks on Web services standards, such as SOAP and WSDL, and MitB attacks. Despite the quick elastic resource provisioning, the DoS attack vector is wide in clouds. A closer look to the table exposes a large set of cloud-specific attacks targeting bandwidth, which is typically under-provisioned in data centers.

4.6 Network security issues

Not only enterprise computing is reshaping, but also the network landscape within an enterprise network. In the past, networks would be static with topologies and servers within lasting for long. Today, networks are something else—a live, dynamic network. The necessity for applications and services connectivity changed the perimeter security. Networking protocols illustrate the change, moving from Routing Information Protocol (RIP) version 1 to dynamic routing protocols like Open Shortest Path First (OSPF) and Cisco proprietary Enhanced Interior Gateway Routing Protocol (EIGRP). Hence, so does the security community needs to adapt to new trends. In the context of network security, those trends are driven by the growth in mobile-based devices and virtualized networking. The following discussions are focused on *mobile platforms* and *perimeter security*.

4.6.1 Mobile platforms

In a way, the adoption of the BYOD paradigm as enterprise norm has also been painful for companies. The unfold of employees using their own devices to access enterprise applications is advantageous from a productivity viewpoint, but that does not hold true for security purposes. Smartphones are increasingly being used to access backend SaaS cloud applications. Not only malware proliferation is increasing in mobile devices [56], but also vulnerabilities. The HP 2012 Cyber Risk Report [113] states that mobile platforms represent a major growth area for vulnerabilities.

Rooting or *jailbreaking* smartphones further enhance the problem because malware can access kernel parts more easily. *Rooting* or *jailbreaking* allow users to install fancier applications by accessing other parts of the operating system, which otherwise would be inaccessible. Hence, a malicious application can reach sensitive components of the operating system, including previously protected decrypted data, which can enable *rootkits* to *escalate privileges* [151]. Moreover, underground distribution channels can redistribute potentially malicious applications because no security is enforced. Applications from such sources are installed at each one own risk. Nevertheless, history proves otherwise as there have been malware distribution across Google Play, a trusted source. It was found to be infected with the Android.Dropdialer [76] *trojan* for months, and a one-click fraud was recently found to scam users into subscribing to a paid service by luring them with adult-related content [106]. Unlike Google Play, Apple App Store is less reluctant to such incidents because of stringent proprietary rules and policies. Li and Clark [151] outlined an attack based on Short Message Service (SMS) that has been around for long. Two types of attacks were discussed. The first has the objective of sending premium-rate SMS messages to offshore accounts or mass SMS spam advertisements with the intent for *phishing*. The second has the objective of using the smartphones to be part of a highly efficient and stealthy *botnet*. Wueest [299] said that mobile spam is gaining ground.

Grispos et al. [99] demonstrated a vulnerability in cloud syncing mobile applications, such as Dropbox. Forensicators could extract logs and retrieve deleted files from those applications because they act like a mirror for what is in the cloud. But the exploit was possible due to a proxy view contained in such applications, which could allow an attacker to gain access to the data stored in the cloud, without accessing it directly. Furthermore, it was discussed in [119] on the fact that data is left behind on mobile phones even after deletion or a factory reset, which can lead to inadvertent *data leakage*. As termed, *phone recycling*, can leak not only private data, but also company data due to the BYOD paradigm. Nearly a third of 500 people who owned a second-hand or refurbished device found remnants of data, according to a survey conducted by BlackBelt and YouGov. Therefore, organizations cannot overstate the continued increase of mobile devices, and it is expected the commensurate rise in mobile vulnerabilities to continue unabated for the foreseeable future [113]. In fact, 2012 saw the emergence of the first documented Android botnet in the wild [56], thereupon corroborating the trend. The static network security approaches are therefore over. A shift from endpoint security to a holistic security approach is required, monitoring and analyzing assets from a higher-level network perspective where data is in motion.

4.6.2 Perimeter security

Traditional perimeter security is composed of static security controls. Network security devices are placed in network traffic aggregation points and on gateways. They are also put in the frontier of the inner perimeter and the outside environment, like DMZs. This approach assumes a fixed network infrastructure, but that is not what is nowadays happening. As discussed previously, the BYOD paradigm is changing the security landscape of the networks and so has been the necessity to open connectivity for services and applications. There are no boundaries [276].

Cloud computing networks, however, are more diverse, dynamic, and mobile. VMs change from one place to another whenever required, and there is a great number of services to be served to the Internet. This means a door per customer must be opened in order for them to access the services.

By default, this is an issue. But, other obstacles regarding the design of cloud networks arise. For example, a firewall maintains a TCP connection table that contains all TCP connections state that the firewall handles—a stateful firewall. Now suppose that a VM beyond the firewall is being accessed externally by a customer. If the VM is migrated to another spot on the cloud which changed the network traffic path, the firewall will eventually timeout the connection and other firewalls that did not knew of the connection might drop outgoing traffic for security purposes. For instance, in the case of a Web server, it is not normal for one to start a connection or be the first one to generate traffic, at least that is what the other firewalls might compute. Worse, given the size of botnets and their power in *flooding* attacks [217], firewalls might just not be able to handle a extremely high number of new incoming connections, in the TCP `SYN` *floods* case, and therein crash. Moreover, the assumption of that the DMZ is the only place where the network is accessed from outside does not hold true for cloud networks [249]. Malware can spread itself from one customer service to others in multi-tenant clouds. In this case, the threat comes form an insider, a nearby tenant. Wu et al. [297] showed that both *sniffing* and *spoofing* attacks could be achieved by exploring the bridge and route modes of Xen, respectively. However, the use of Security Virtual Appliances (SVAs) on hosts [26], rather than on the perimeter, allows to introspect traffic in and out of VMs, therefore preventing such attacks [26]. SVAs are what vendors now offer in their state-of-the-art security solutions. Rather than physical appliances, SVAs come as virtual blades that can be added as needed, hence being scaled as required and supporting the cloud business model. Amazon EC2, for example, provides a firewall solution to each customer. A mandatory inbound firewall is by default configured in deny all mode and the customer must configure a port to allow incoming traffic. This traffic may be restricted by protocol, by service port, or by IP address [9,27,162]. Nonetheless, the boundary threshold in firewalls to accept new incoming connections might pose another issue.

The big challenge on the cloud provider side is to achieve the desired security level as one would in standard enterprise networks. This calls for monitoring and logging events. However, Grobauer et al. [100] stated that standards and control mechanisms are still scarce for cloud networks. Furthermore, log files record all tenants events, which may hamper or impede one to prune for a single tenant due to *insufficient logging and monitoring capabilities*.

4.6.3 Summary

The network category discusses the shift of enterprise networking throughout the time, but it particularly focuses on illustrating its security issues by underpinning nowadays trends. Those security issues previously overviewed are condensed in Table 9. The proliferation of smartphones and portable computing devices allowed the emergence of the newly BYOD paradigm. This brings new classes of security issues that have an intrinsic relationship with mobile-enabled malware spread and an impact in cloud applications accessible though mobile platforms. Furthermore, cloud computing changes the customer enterprise perimeter borders, like DMZ positioning, but it also opens the providers real and virtualized networks in terms of connectivity.

4.7 Access security issues

It is common to see online resources being protected in terms of authentication with an email or username and password combination. Cloud environments adopt this approach, and because frontend interfaces are built with Web technology as any other Web site, the problems related with access are also

Table 9 Summary of the security issues and respective studies regarding the network category of the taxonomy

Category	Topic	Issues	Studies
Network	Mobile platforms	Mobile malware growth	[56]
		Mobile vulnerabilities growth	[113]
		Rooting and jailbreaking, rootkits, privilege escalation	[151]
		Untrusted underground application distribution channels	–
		Cloud syncing mobile applications vulnerabilities	[99]
	Perimeter security	Static network infrastructures	–
		Boundary-less network perimeter	[276]
		DMZ assumption	[249]
		Firewalls new incoming connections limit	[217]
		VMM network sniffing and spoofing	[297]
		Insufficient logging and monitoring capabilities	[100]

relevant for such systems—the Web is an attack vector [295]. Multi-tenant clouds have a great number of customers accessing their own resources. It is important to logically and physically segregate resources from one another. In turn, it is also key to deploy security policies that address those issues in terms of authentication and authorization requirements [35]. SaaS applications must support being customizable and configurable to incorporate specific access conditions. The discussion will now evolve to the subject of security issues related with *physical access*, *credentials*, *authentication*, *authorization*, *Identity Management*, and *anonymization*.

4.7.1 Physical access

Data centers centralize massive amounts of data in a single point. Because this data is not owned by the provider, but by a diverse number of heterogeneous customers that outsource their businesses, it can be ambitious in terms of profit to somehow retrieve any kind of useful sensitive information. Due to the security issues that have been discussed throughout this article, it is crucial to guarantee physical security in order to prevent any kind of *data leakage* by exploring the wide cloud attack vector from an inside position. After all, there is good reasons to have so much aspects into consideration when building data centers, like the ones discussed in Sect. 3.4. The ultimate goal is to protect information in long-term in a co-habitat like cloud environments.

The appealing side of clouds can attract a dangerous community from the outside, but also from the inside. *Malicious insiders*, as are usually called [3,203,306], can overtake the physical security controls and penetrate the facilities. Unpleasant or ex-employees, hobbyist hackers, espionage agents, or other malicious cybernetic actors, can patiently wait for the most opportunistic moment to attack, either from outside or from the inside. In fact, cybercrime now relies on what are called *money mules*. These are hired online for the sole purpose of transferring illegally acquired money to other bank accounts typically offshore, rendering the money untraceable. However, these *money mules* are not aware of the of illegality and think the employment is legitimate. Here [236] is one example. The same concept can be applied to data center facilities and cloud environments, by impersonating a character, say a customer, and instigate a way in.

Outside threats pose greater impact on clouds, not only in terms of system damage, but also to the provider reputation and business. due to the long-term loss of leaving customers [28]. Nevertheless, a single incident from a *malicious insider* can have leak a great amount of data. *Malicious sysadmins* can install all sorts of software and access VMs [242]. For example, XenAccess allows to run a user level process in Domain0 that directly accesses VMs memory contents at run time. With physical access, other more sophisticated attacks can be accomplished, such as *cold boot* attacks and *hardware tampering*. Therefore, monitoring of privileged *sysadmins* with malicious intents should be carried along with their accesses controlled [132]. Zou and Zhang [313] further discussed that a *malicious insider* can remove security-specific kernel modules, such as firewalls and anti-viruses, making systems purposely vulnerable. Henceforth, deploying perimeter security along with Access Control Lists (ACLs) and is mandatory.

4.7.2 Credentials

Usually, the Lightweight Directory Access Protocol (LDAP) or the Microsoft Active Directory (AD) technologies are used in large companies to manage access credentials and for authentication purposes. In the cloud computing paradigm, LDAP and AD servers can also be outsourced, placed in systems of the cloud provider, or within the company network, behind a firewall. The first option increases IT *management overhead* if multiple applications are deployed, because it is required to add, modify, disable, or remove accounts every time employees leave or enter the company [261]. In addition, the *loss of control* issue also applies herein, specifically in terms of losing control over the configurations and security of LDAP or AD servers. Grobauer et al. [100] recalled past weak password-recovery mechanisms to exemplify the *weak credential-reset* vulnerability at the provider side, when in charge of managing credentials.

Computer experts always had to deal with security issues affecting credentials, because they pose dangerous menace if stolen by means of, for example, *phishing*, *keyloggers* or by establishing MitM attacks [28]. If credentials are compromised, it is possible to monitor or manipulate data and transactions, along with performing *malicious redirects*. Therefore, *replay sessions* are most likely to happen. Furthermore, it is also possible to deploy DoS attacks by using legitimate accounts for hiding the identity of the attackers. Finally, User to Root (U2R) attacks may allow gaining root level access to VMs or hosts through a valid user account [171]. Making a concealed attack base from compromised accounts sounds even worse. In this case, perimeter security has already been surpassed, but there is yet more security layers to overcome. Nevertheless, it is already possible to cause *service disruptions* or *business halts*, in turn leading to *customers loss* and *financial loss*.

4.7.3 Authentication

Chow et al. [50] argued that requiring authentication prior to providing access to SaaS applications is advantageous because of centralized monitoring, which makes software piracy more difficult. Because most remote authentication mechanisms rely on regular accounts, they are nevertheless

susceptible to a plethora of attacks [94], such as *brute-force* and *dictionary* attacks. Common approaches available nowadays include simple text passwords, third-party authentication, graphical passwords, biometric scans, and 3D password objects [73]. Simple text passwords are perhaps the most commonly used mechanism, but Hart [107] said that *archaic static password*—one-tier login—is now simply not enough, as it constitutes one of the biggest security risks. Third-party authentication is not preferred for smaller cloud deployments either [73]. Graphical password schemes have the disadvantage of requiring more user time, while biometric approaches, such as fingerprinting, palm printing, and iris or retina recognition, require physical presence, therefore being adequate to be part of data center security, and not always applicable to remote authentication. Finally, approaches with 3D passwords do not support multi-level authentication.

Cloud customers are most likely to subscribe multiple services, resulting in multiple login requirements. In addition to being difficult implementing strong authentication at the user level [281], it is complex to manage and create multi-level authentication mechanisms for several services. Single Sign-On (SSO) techniques address these issues. Google, for instance, was once vulnerable in their Security Assertion Markup Language (SAML)—as defined by the Organization for the Advancement of Structured Information Standards (OASIS), a consortium for open standards—implementation for SSO [158]. SAML allows to exchange authentication and authorization information between two parties, such as an Identity Provider (IdP) and a service provider. The Google implementation shared XML-based authentication data across multiple servers, allowing a user to switch between services running on different servers, like Gmail and Calendar, without re-authenticating. It was possible to capture and use SAML data to carry *impersonation* attacks. This loophole was nonetheless closed. Somorovsky et al. [254] found eleven SAML frameworks to be vulnerable to XML *wrapping* attacks from a total of fourteen frameworks, including Salesforce, Apache Axis2, and OpenSAML, to name a few. The vulnerabilities can be exploited using few resources, and because SSO systems may become a single point of authentication, this study raises alarming concerns for cloud environments and authentication in general.

Traditionally, authentication methods have backup recovery schemes to allow resetting the password given enough proof of the account ownership. This is the case of Questions and Answers (Q&A). Questions are asked at account registration time, to which the user answers. If the scheme is used, those same questions are required to be answered correctly. However, there are a finite number of answers, and a little bit of *doxing* can help get on the right path. Users may also not memorize the answers in long term or misplace them if they are written down to a piece of paper, for example. In addition, each implementation is usually different, having distinct questions, thus rendering answers difficult to manage. Random answering is a stronger approach and might solve the latest issue, but increases the memorization complexity. Google finds it probably better to abandon the Q&A approach [101]. In fact, it provides an SMS-based recovery system. Authentication methods incur in one more issue. It is possible to provoke an *account lockout* state, a form of DoS [100], by exploring the threshold for the number of login attempts often used by authentication mechanisms. It is possible to repeatedly, and in quick succession, try to login with a valid username until the limit is achieved.

4.7.4 Authorization

Centralized access control could be advantageous and ease several management and security tasks. However, that may not be possible or desirable in a scenario populated with mashups of data, which are most likely to be seen in the future [50]. The development of data mashups have security implications in terms of *data leakage* and on the number of sources a user retrieves data from. The deployment of access authorization mechanisms for each data source has potentially prohibitive implications in terms of usability. For instance, Facebook does not typically verify third-party applications that use data uploaded to Facebook servers. *Malicious applications* can, therefore, perform malicious activities. Other social sites are also affected by similar problems [294]. Authorizing third-party applications to access certain private information is dangerous. For example, one can authorize outside applications to access cloud-based hosted applications. Social networks widely implement such mechanisms and, for the malicious actor, it is easier to acquire intelligence on targets. This widens the risk scenario. An attacker can either build a *phishing* attack more easily or somehow profile the underlying cloud system.

Grobauer et al. [100] identified *insufficient or faulty authorization checks* as possible exploitable vectors. The authors exemplified with an *insecure direct object reference*, an issue placed forth in the 2010 and 2013 top ten Web application security issues of the OWASP [200,201], called URL-*guessing* attacks in their study. Service management interfaces are also prone to offering *coarse authorization control models*, making it harder to implement duty separation capabilities.

4.7.5 Identity management

Identity Management (idM) is a broad administrative area that deals with identifying entities (e.g., individuals or enterprises) and cloud objects, controlling access to resources according to pre-established policies [175]. Subashini and Kavitha [261] provided idM in three perspectives: the pure identity, log-on, and service paradigms. The first perspec-

tive manages identities with no regard to access or entitlements. The second perspective concerns the traditional methods using physical tokens, such as smartcards. The third perspective delivers online, on-demand, presence-based services with respect to roles, appropriate to cloud services. Moreover, three idM supporting models were identified, namely the independent idM stack, credential synchronization, and the federated idM. An independent idM stack is maintained at the provider end, keeping usernames, passwords, and all related information per SaaS application. This model should be highly configurable to comply with the customer security policies. Synchronized credentials consist in replicating account information to the provider end stored at the customer end, giving access control abilities to the provider. *Account information leakage* is the main threat in this model, both in storage and in transit when replicating the data to the provider. Federated idM is the means of linking account information stored across multiple idM systems, being SSO a feature of this model. The authentication occurs at the customer end, while users identity and certain attributes are propagated on-demand to the provider using federation. This model has to cope with *trust* and *validation* issues. PaaS and SaaS platforms have complex hierarchies and fine-grained access capabilities, raising logistic and transport issues when synchronizing data [248]. Takabi et al. [270] also discriminated an interoperability issue that could result from using different identity tokens and identity negotiation protocols.

4.7.6 Anonymization

One particular technique to prevent association of data to an entity is to use anonymization. This cuts the semantic links of the data to their owners while preserving the provider capability of charging for resource usage in a proper and reliable manner [128]. Therefore, it is another layer of security that is implemented right into the database. Actually, enterprises have felt increasing pressure to anonymize their data until proper privacy measures are in place [50]. A few techniques to anonymize data in clouds have been provided [31,128]. Anonymization is, nonetheless, a hard task to complete [50], and even more when a few threats and attacks are incurred.

Xiao and Xiao [300] discussed the *hidden identity of adversaries* threat. Identity information of cloud customers should not be disclosed due to privacy requirements, which is the reason why some systems implement anonymous access techniques. However, full anonymity requires all the information to be somehow hidden. Therefore, malicious actors can jeopardize the security state because it is easier to be undetectable. Moreover, a class of *de-anonymization* attacks has been a particular research topic. Backstrom et al. [23] proposed a family of attacks—the *active*, the *passive*, and the *semi-passive* attacks—which *breaches edge privacy* of a targeted group of individuals on a social network with basis on structural knowledge. Narayanan and Shmatikov [179] proposed an algorithm with only a 12 % *de-anonymization* error rate on an online photo-sharing Web site purely based on network topology information. Moreover, Ding et al. [74] spearheaded *de-anonymization* attacks on dynamic social networks by using correlations between sequential releases. A curious case resulted in identifying the governor, at the time, of the Massachusetts state in the USA by de-anonymizing health records. It was proved that *innocuous and neutral data injection*, like gender and birth date information, on anonymized data can lead to the identification of the entities [248].

4.7.7 Summary

The connectivity openness discussed in a previous section is thwarted by an ample group of access security issues, starting from small granular served assets to big outsourced network structures. The access category, whose security issues are summarized in Table 10, should be analyzed from two perspectives: the insider and the outsider issues perspectives. From the inside to the outside, the danger comes from the possibility to physically eavesdrop data or from information disclosure. From the outside to the inside perspective, the issues move to the topics of authentication and authorization. Authentication methods based on common credentials can be prone to theft or breaking, while LDAP or AD servers can either be placed on-premises or off-premises, with each option bringing different obstacles. The inapplicability of alternative authentication approaches and the degradation of security controls like Q&A and login thresholds calls for new mechanisms. The new SAML standard has also shown vulnerabilities in this regard. Social networking and data mashups expansion added yet another attack vector to the portfolio of Web security issues, entailing malicious third-party applications and vulnerable authorization models. Clouds also encompass idM and federation issues, and face de-anonymization attacks.

4.8 Trust security issues

For customers to outsource their businesses and data, trust must be put on the cloud provider and on the off-site locations. Not only that, but also the other way around, cloud providers must trust customers to access their clouds in an supposedly honest manner. In addition to the cloud stakeholders, the trust is also related with the assets in them, including computational algorithms, storage hardware, virtualization techniques, and Web-based access [131].

Trust alone, which sometimes cannot be established, may not be enough to make cloud customers comfortable [128, 313]. Thus, additional means are expected to exist in order to boost customers confidence. Firdhous et al. [83] added

Table 10 Summary of the security issues and respective studies regarding the access category of the taxonomy

Category	Topic	Issues	Studies
Access	Physical access	Malicious insiders	[306,313]
		Malicious sysadmins	[132,242]
		Cold boot attacks, hardware tampering	[242]
	Credentials	LDAP and AD servers location, IT management overhead	[261]
		Weak credential-reset vulnerability	[100]
		Phishing, keyloggers, MitM attacks, malicious redirects, replay sessions	[28]
		U2R attacks	[171]
	Authentication	Archaic static password	[107]
		Inapplicability of alternative password schemes	[73]
		SAML vulnerabilities	[158]
		XML SAML wrapping attacks	[254]
		QA vulnerabilities	[101]
		Account lockout	[100]
	Authorization	Centralized access control inapplicability, data mashups	[50]
		Malicious third-party applications	[50,294]
		Insufficient or faulty authorization checks, frontend interfaces coarse authorization control models	[100]
		URL-guessing attacks	[100,200,201]
	Identity management	Synchronization leakage, federated idM trust and validation	[261]
		Complex and fine-grained synchronization	[248]
		Distinct identity tokens and negotiation protocols	[270]
	Anonymization	Hidden identity of adversaries	[300]
		De-anonymization attacks	[23,74,179]

that trust management plays a vital role, not only in cloud systems, but also in other distributed systems, Peer-to-Peer (P2P), and sensor networks. Below is included a discussion of security issues related with *moving to the cloud*, the *human factor*, *reputation*, *auditability*, and *anonymization*.

4.8.1 Moving to the cloud

Amoroso [13] provided a study on the current trust state of enterprise perimeter model. The typical security perimeter is composed of a static closed network with restricted connectivity for business-related applications and services. As technologies matured, including the Internet, the trust in such a perimeter decayed throughout the years. The diversity of communication alternatives created business dynamism, but it also opened doors for perimeter breaching. That initial static approach was a more trusted model because IT at the time was mainly used for email service and occasional Web access. Later on, in the late nineties, the widely adopted VPN technology hammered down yet again the trust level by allowing remotely connected employees access internal assets from external locations. Moreover, if an enterprise A trusts enterprise B that, in turn, trusts C, then A trusts C, therefore creating a simple transitivity [262], which lowered the trust level once more. Next, the increased Internet traffic and connectivity exceptions for enterprise applications and services allowed for more dangerous security threats to appear, like malicious Web pages hosting malware. Finally, the latest pounding factors are APTs and mobile devices that can easily hop from internal Wi-Fi networks to radio-based carrier broadband networks (e.g., 2G, 3G, and 4G). Putting it all together, a nullified trust model is rendered.

As discussed above, trust on enterprise perimeter security has broken down to pose a serious threat. Companies are mainly targeted for *exfiltration* even though a more restrictive network is in place. The point is, if such holds true, then a cloud provider network environment can be abysmally open for allowing connectivity to an umpteen number of applications, services, and tenants for customers from all around the world. As discussed throughout this article, this raises issues with regard to storage, computa-

tion, and access to cloud instances, namely *malicious insiders* [79]. Ultimately, trust is pushed farther back into the machines rather than staying at an holistic view level of a network, and SVAs placement within VMMs are proof of it.

4.8.2 Human factor

Humans are the root for all problems. Humans design and build to fit their own needs, but then the result is faulty in some aspect and humans strive to fill the gaps. There is still a reliance on perimeter defenses when the reality is that are no boundaries [276], and the cloud computing model is proof as data is moved around from server to server. Also important, enterprises must trust employees in order for them to carry out their duties—*you cannot firewall human nature*. Nonetheless, that may require access to a load of assets. In a data center, there are a great number of production assets that need to be actively maintained. That leads to a problem because human error or negligence is most likely to happen.

Both cloud users and system administrators should have a particular care for their password choices. In a big ISP or cloud provider, tens of thousands of physical and virtual servers need to be managed. Saving strong passwords for every single node and remembering them or storing them in some encrypted database (e.g., KeePass) might be hard or even impossible. Moreover, most employees share their passwords with a coworker, a friend, or even a friend of a coworker, even after receiving specific training [279]. Furthermore, it was showed that the word *password* is the most common password used in the world. The study was performed by SplashData, a company dedicated to address password concerns in IT, in the Worst Passwords of 2012 report [75]. The study was compiled with millions of passwords published online by hackers.

Another big problem related with the human factor is called *social engineering*—humans are the weakest link of computer systems. Reporter Mat Honan got badly hacked [112] when someone called both Amazon and Apple support. Because Amazon had a deficient password-recovery policy, anyone with the name of the account, associated username and billing address could call Amazon to input a new credit card. Then, Amazon could be redialed to add a new email address to the account, to which the previous information and the newly inserted credit card number are required. To get into the Apple account, the last four digits of the real credit card information shown on Amazon account were transmitted to Apple support, who immediately issued a temporary password.

Typical *social engineering* tactics used in massive *spam* campaigns include utilizing spoofed brands, mainly related with the drug industry [56]. In addition, traditional cybercrime takes advantage of occasional or single events to send out *spam* waves. Examples include releasing of new smartphones, operating systems, or during the tax season. More recently, the death of Margaret Thatcher has been quickly introduced into the BlackHole exploit kit for *phishing purposes* [61]. *Spear-phishing*, on the other hand, is more focused and, therefore, utilizes techniques that recur to terms not strange to targets. Such terms include common business terms [84].

As Thompson [276] discussed, the interpretation of a *social engineering* situation varies from person to person. If a security expert is confronted with a *phishing* email, the subjective analysis it performs with basis on past experiences and knowledge is key toward the interpretation of the email. However, a common Internet user with no security awareness is easily phished. It all comes down to the decision of an single quick moment: clicking a link or opening a file.

4.8.3 Reputation

If various VMs of different customers are hosted by the same machine, they share the same hardware assets. Thus, activities and behaviors of cloud stakeholders affect each others reputation, an issue known as *reputation isolation* [82,174] or *fate-sharing* [47,229]. If a cloud system is subverted, the users using the system may be affected and their services disrupted. Additionally, all of them benefit from the security expertise concentration that the cloud offers, depending on the signed SLAs, consequently sharing the same infrastructure and fate. For instance, in 2009, Amazon EC2 was subverted by spammers, causing blacklisting of a large number of IP addresses belonging to EC2, in turn provoking major service disruptions [47]. A second noteworthy incident occurred in the same year, in which federal agents seized data centers suspicious of facilitating cybercrime. Many cloud customers, namely companies, without knowledge of the criminal activities, faced disruptions or complete closures of their business.

4.8.4 Auditability

Assessing the health status of the assets in cloud environments is hard for customers and third-party auditors because cloud providers may not be willing to provide metadata information on the outsourced infrastructures. Auditability consists in performing a series of tests to find out if all proper implementations are in conformity. In cloud environments, an additional layer above virtualized guest OSes would also allow that [17]. For instance, a company retention policy might require the *provability of data deletion* when outsourcing data [50], thus being indispensable to check for proper data deletion enforcement on cloud systems.

Audit techniques analyze service conditions, monitor intrusions, accesses and other events, and record logs with a detailed description of what happens are suitable, according to [94], for assuring that security measures are employed. According to the same authors, trusting only the reports or evidences of the providers is not enough. Nevertheless, providers might not be willing to allow auditing tasks [82]. Customers can delegate audit responsibilities to trusted specialized third-party auditors, reducing their burden on this aspect. Moreover, auditability eases the process of identifying the responsible party in case of a legal action, which can be vital to the cloud stakeholders, since it helps limiting the scope of search and seizure of electronic data, while assuring that law enforcement agencies do not overreach when carrying out their duties [47]. In fact, data might be forced to be kept within jurisdictional bounds so as to make auditability comply with the law perspective. Consequently, some businesses may not like the ability of agencies to obtain their data via the *court system* [310]. Even if all these barriers were surpassed, auditability tasks still have to endure against the *data locality* issue and to the fact that some techniques are not privacy-preserving capable, motivating research activities in this area [229].

4.8.5 Anonymization

Google, in the search marketplace, modifies the last IP address byte after 9 months of a specific search, and deletes it after 18 months. It also anonymizes cookie information with a process called generalization, which can guarantee a reasonable level of privacy [278]. Anonymized data is nonetheless stored for internal purposes. This example shows how enterprises handle user usage information, like IP addresses, to tune up their algorithms or other business products. Of course, this approach may not be well received for some who think privacy matters are at stake here. So, the same is applicable to cloud environments. Customers require to trust their cloud providers security control logs produced by perimeter security devices so that are not tied to a particular business or customer, hence the anonymization. This requirement provides an additional layer of security so as to prevent infer some information based on the logs, in turn safeguarding, for instance, VM location of particular customers against malicious insiders.

4.8.6 Summary

Table 11 summarizes the cloud security issues of the trust category. Trust refers not only to the providers trustworthiness in compliance and honest matters, but also to the very infrastructure and the human factor. The latter is mostly associated with a faulty security awareness and training of employees and users in general, resulting in large amounts of phishing campaigns, aimed at applying social engineering techniques and at weak password choices. The move to the cloud may have a negative impact to business due to the aforementioned issues and to the openness of network infrastructures, thus potentially affecting other cloud tenants as well. To mitigate trust problems, customers are urged to audit their assets, but even audit mechanisms are hard to implement.

4.9 Compliance and legality issues

The cloud business model uses SLAs to specify the agreements over a certain service, may that be in the form of IaaS,

Table 11 Summary of the security issues and respective studies regarding the trust category of the taxonomy

Category	Topic	Issues	Studies
Trust	Moving to the cloud	Enterprise nullified trust model	[13]
		Cloud environments openness	–
	Human factor	Employees trustworthiness	–
		Password sharing	[279]
		Password commonness and strength	[75]
		Social engineering	–
		Phishing and spear-phishing	[84,276]
	Reputation	Reputation isolation	[82,174]
		Fate-sharing	[47,229]
	Auditability	Providers willingness in providing status information	–
		Providers reports trustworthiness	[94]
		Jurisdictional audits, court systems	[47,310]
		Data locality	–
		Lack of privacy-capable audit techniques	[229]
	Anonymization	Logs anonymization	–

PaaS or SaaS. An SLA is always signed to formally agree on a price per service and inherent legal matters. Thus, there can be an implied subjectivity on the fulfillment of such agreements. *Forensics*, *acts*, *legal problems*, *accountability*, and *governance* will now be discussed throughout the next sub-subsections.

4.9.1 Forensics

Computer forensics, or digital forensics, is a particular form of auditing that has emerged in recent years to fight cybercrime. The development of this field has been motivated by the interest of organizations in audit tasks. It has the objective of determining potential digital evidence by means of analysis techniques [274]. When applied to clouds, digital forensics face a complex scenario because data is pushed further back into the network and servers and is more spread out across them, rather than purely being on a physical computing device. Forensics also face the *data locality* issues, making it hard to isolate particular resources. Private clouds, nonetheless, are easier to deal with when compared to public ones, since servers, applications, databases and other resources are easier to identify [274].

From the user perspective, forensics present concerns of *data seizing* and *data disclosure*, compromising confidentiality and privacy; while from the forensicators perspective, the cloud stack presents different issues. Key evidences may reside simultaneously inside and outside the cloud, as for example in the Web browsers history and caches [52]. Additionally, the BYOD paradigm might also bring difficulties in the sense of getting legal authority to investigate a user personal device which falls outside the enterprise reach. In addition, cross-platform SaaS applications might also present obstacles in terms of appropriate and applicable techniques that work with a variety of devices, like smartphones [274]. This results in added difficulties for data *collection*, *collation* and *verification*. Taylor et al. [274] added that the hardest aspect of cloud storage investigation is to find out what a user did, from the beginning to the end of a service subscription. Nevertheless, computer forensic techniques can be employed to obtain complete history of the VM, including usernames, passwords, applications, services, Internet browsing history, IP addresses, and protocols that connected to the VM [79]. Gonzalez et al. [94] named the issue of *hardware confiscation*, a result from applying law enforcement, to *e-discovery*, saying that *data disclosure* is a critical issue in these cases.

Another problem is that virtualized environments may produce *unsound forensic data*, as current forensic techniques are not adequate for the IaaS model [52]. Moreover, the application of some security measures impact forensic activities negatively, since investigators have to handle encryption schemes, privacy protecting acts, and time-consuming procedures to gain legal authority to investigate cloud infrastructures. In contrast, as previously discussed, some cloud environments might not provide such cryptographic mechanisms due to computational overhead, thus rendering a *lack of validation for disk images* [80]. Forensic practitioners heavily rely on hash values to compare disk states, and if that information is nonexistent before a crime, then examiners and jurors are unlikely to accept the evidence presented. Several studies [80,274] pointed out that *evidence acquisition* is a forefront security issue in cloud forensics.

Dykstra and Sherman [80] emphasized the importance of layers of trust in cloud environments, as the jury or judge of a legal action ultimately has to decide whether or not the evidence presented is believable, reliable, and trustworthy enough. In their study, the IaaS model was divided into six layers, which are network, hardware, host OS, virtualization, guest OS, and services, sorted upwardly. In each one, different kinds of trust and forensic activities are required. In private clouds, the cumulative trust decreases as one moves from the top to the lower layer. In public clouds, however, trust is needed in all layers, especially to deal with *malicious insiders*.

4.9.2 Acts

Given that cloud computing is a relatively new technology, the current cyberlaws do not yet cover the requirements posed by it. From the cloud customer point of view, the privacy of its data is at peril because of outdated law acts. In addition, acts form different countries do not hold consistent among them, which might create a conflict point when data travels across borders, as discussed in the following sub-subsection. For instance, the USA PATRIOT Act (UPA) conflicts with the Personal Information Protection and Electronic Documents Act (PIPEDA) in Canada and with the Data Protection Directive in Europe. Other examples include older regulation acts, which fail to protect individual privacy and business secrets, being out-of-date and inapplicable to new cloud scenarios involving three stakeholders. The Electronic Communications Privacy Act (ECPA) of 1986 and the UPA of 2001 are cited as examples of acts that fail to protect data being disclosed to government entities. The Fair Credit Reporting Act (FCRA) of 1970, the Cable Communications Act (CCA) of 1984, the Video Privacy Protection Act (VPPA) of 1988, the Health Insurance Portability and Accountability Act (HIPAA) of 1996, and, finally, the Gramm-Leach-Bliley Act (GLBA) of 1999 are cited as examples that fail to protect data being disclosed to private parties [310]. Furthermore, laws may oblige providers to examine data contents for evidence of criminal activities and other government security matters. This is the case of recent, although some were already rejected, proposed acts: the

Stop Online Piracy Act (SOPA), the Preventing Real Online Threats to Economic Creativity and Theft of Intellectual Property Act (PROTECT IP Act), the Anti-Counterfeiting Trade Agreement (ACTA), and the Cyber Intelligence Sharing and Protection Act (CISPA). Such acts can breach the privacy of any customer in any enterprise in any provider by disclosing private information to certified government parties as long as it falls under cybersecurity purposes. This certainly would have impact on cloud businesses because cloud providers would have to legally hand over data to the government.

4.9.3 Legal problems

The cloud operation raises many compliance and legal issues. The most prominent is the *multi-location* particular characteristic of cloud environments. Cloud providers with enough resources have data center facilities spread all over the world, allowing them to publicize high availability and geographic redundancy characteristics in marketing campaigns. Regardless of that, some countries do not allow data to leave its boundaries. In this scenario, several issues arise. If data flows across borders, it cannot be determined under which country jurisdiction the data falls. If an incident takes places, it is hard to say to which extent legal authorities can reach in order to find responsible parties. This includes assessing whether government agencies can access the information outside the borders of the country in which data was generated in the first place. Moreover, it is hard to say whether the governments of a country hosting a data center are entitled to peek into data generated elsewhere. Finally, a costumer may face a serious problem when served by a subpoena or other legal action under a limited time-frame, since it may not be possible for the provider to gather the necessary answers and results within the time-frame. These problems are yet to be fully dissolved [50,135,174,211].

Service Level Agreements (SLAs) are agreed with basis on the premise that the specified requirements are respected throughout the entire duration of the contract. *Dishonest computation*, *accidental resource allocation*, *availability issues* and *data loss* are, nonetheless, problems that can violate SLAs [137,300]. To determine the cause of such problems is hard for both customers and providers, raising compliance and legal issues. Thus, the risk of investing in certification is high to customers, since providers can fail to provide compliance evidences [82]. In certain cases, using public clouds implies that certain kinds of compliance cannot be achieved, such as the data security requirement PCI Data Security Standard (DSS) [209].

Other compliance and legal issues are related with differently aligned interests between cloud stakeholders [50]. *Limited usability*, *implied*, and *obliged contractual* or *unclear terms* can pose issues in the service usage context. Once an SLA is closed, the customer remains at the mercy of the provider. As a result, customers may trust a certain provider more or less with basis on the SLA it offers. However, SLAs are not consistent among providers, creating obstacles in identifying trustworthy providers [104]. An issue named *transitive nature* by Chow et al. [50] was also described by Pearson [211], although named dynamic provisioning by the latest author. Both terms refer to the usage of subcontractors by providers, in which case customers have even less control, influence, compliance certainty, and trust. In such case, it is potentially more difficult to find the responsible party when an incident happens. Problems of this type have happened in the past [50], resulting in *data loss*. More issues to customers arise when providers must obey government orders in disclosing data of lawful interception [174]. When permitted, this kind of actions might break the chain of trust created with providers.

4.9.4 Accountability

The pay-as-you-go cloud business model allows customers to rent bandwidth and resource usage. Due to the extravagant resources used to fight back *flooding* or *resource exhaustion* attacks, billing can be drastically raised to customers running the targeted services, at least if the attacker cannot be identified [129]. Even if it can, QoS properties can drop even when the hosting servers can sustain such attacks. Xiao and Xiao [300] justified the previous statement with the fact that SLAs were signed to some extent of responsibility, meaning that responsible parties must be determined in case of an incident.

A more subtle and evasive attack called Fraudulent Resource Consumption (FRC) [117,118] is a form of an Economic Denial of Sustainability (EDoS) attack [300]. It explores the pricing model to harass billings, having the purpose of causing *financial loss* to the victims. Attackers seem legit users who continuously, and for a long time, send requests in order to consume bandwidth, but not enough to cause a DoS. FRC traffic is hard to analyze and classify, hence raising its severity. Besides bandwidth accounting, there is also the problems of storage and computing accounting. In order to fulfill the measured service in clouds, servers must be correctly accountable. As Aguiar et al. put it, an accountable system should take into consideration three properties: identity binding, tamper-evident logs, and execution verification. However, as discussed in Sect. 4.3.2, *unreliable computing* or peer entities that do not follow the agreed SLA protocol can promote the accountability system wrongfully. Moreover, cloud providers multiplex applications belonging to different customers in order to achieve high utilization. Nonetheless, incorrect resource consumption metering may happen, resulting in *inaccurate billing*, possibly giving addi-

tional costs to customers [247]. So, for the customer viewpoint, it can be hard to known it the bill is correct and according to the real usage of the services.

4.9.5 Governance

Governance issues refer to losing administrative, operational and security controls over systems [94]. The *vendor lock-in* issue is particularly relevant in this topic. Interoperability between clouds still faces security and standardization issues, namely concerning protocols, data formats and APIs. As a result, customers might become trapped to a certain cloud provider that outsources their infrastructures, becoming vulnerable to data migration, price increases, reliability and security problems, service termination, or even to the possibility of providers going out of business [17,94]. For instance, standardized APIs would allow customers to deploy SaaS applications and have copies of the same data across multiple providers, mitigating the danger of one cloud provider taking all data copies of customers in case of complete closure. Armbrust et al. [17] said some might argue that, in such case, *race-to-the-bottom* of cloud pricing would flatten the profits of providers. Nonetheless, they present two arguments against such statement: first, customers may not necessarily adopt low-cost services since QoS and security properties do matter, and second, new possibilities to integrate hybrid clouds both on-premises and off-premises would arise.

4.9.6 Summary

In the last category of the taxonomy, the focus escapes from the cloud itself and enters a more subjective topic. The security issues of the compliance and legality category are summed up in Table 12. Computer forensics gained respectable ground, but now have to cope with a great number of obstacles, and part of them also affect the remainder cloud stakeholders. Such issues extend to VMs, mobile devices, data location and to legal contexts. From the cloud customer point of view, current law acts and SLAs enforcement are not

Table 12 Summary of the security issues and respective studies regarding the compliance and legality category of the taxonomy

Category	Topic	Issues	Studies
Compliance and legality	Forensics	Public clouds, data locality, data scattering through servers, legal authority, cross-platform forensic techniques	[274]
		Data collection, collation and verification	[52,274]
		Data seizing and disclosure, hardware confiscation, e-discovery	[94,274]
		Forensic data unsoundness rendering due to virtualization	[52]
		Encryption schemes, lack of validation for disk images, evidence reliability for jurors	[80]
		Evidence acquisition	[80,274]
	Acts	Outdated acts	[310]
		Privacy breaking acts	–
	Legal problems	Data jurisdictional borders	[50,135,174,211]
		SLA violation	[137,300]
		Providers compliance evidences	[82]
		Providers and customers differently aligned interests, transitive nature	[50,211]
		SLAs consistency and trustworthiness	[104]
		Data lawful interception	[174]
	Accountability	Under-attack QoS properties	[129]
		FRC and EDoS attacks	[117,118,300]
		Unreliable computing, protocol violation	[2]
		Inaccurate billing	[247]
	Governance	Vendor lock-in	[82]
		Data migration, price increases, reliability and security problems, service termination, providers business termination	[17,94]
		Race-to-the-bottom	[17]

appropriate to ensure an untroubled cloud service subscription. In addition, ensuring an error-free, scalable, and fine-grained measurable accountability system is not yet achievable, and the EDoS attacks make use of that fact. Furthermore, the lack of standardization implies vendor lock-in and development delays on interclouds, but then fully operational interoperable cloud environments might bring business flattening.

5 Open challenges and recommendations

This section provides a high-level panorama of the security state on cloud environments. It starts by first outlining the main lessons learned from this study, while citing along research highlights on the field. Then, a few practical recommendations in terms of security for both cloud customers and cloud providers are handed out. The section ends with an outline of ideal cloud environments with regard to security.

5.1 Lessons learned and research highlights

Cloud computing is definitely an attractive technology and is capable of delivering on-the-fly extraordinary capabilities in the form of measurable services. The inherent business model allows for enterprises to monetize their businesses, saving costs and raising productivity and profits. Clouds will surely continue to rise and the IT industry will heavily rely on it for supporting enterprise computing and the IoE. The following discussions will focus on underlining the main lessons learned from this work, providing also a few worthy references on the subject.

5.1.1 The contrast between public and private clouds

The shift to clouds still comprises a difficult decision. Several security issues provide good reasons for some not to move their data to cloud environments, especially public clouds. The Alert Logic State on Cloud Security Report [6] depicts the top three incident occurrence as Web application attacks, brute-force attacks and vulnerability scanning in cloud environments. There is lower threat diversity in clouds than in enterprise data centers, meaning that are more different types of threats in enterprise networks. That is mainly due to the APT threats that target private companies for espionage. Thus, on one hand, it is safer to opt for private cloud solutions, like Nebula One [181], but on the other, it is more probable to see sophisticated attacks on enterprise networks. In addition, private clouds bring higher costs, in part defeating the purpose of the utility-based cloud business model. In either case, cloud users still have to cope with issues of the software, storage and computing, virtualization, network and access categories of the taxonomy. From the Internet and services category, issues of Web services and Web technologies also apply in this case, and the human factor should never be set aside when discussing security—security is not a technical solution alone.

5.1.2 Cloud storage and computing

Outsourcing storage and computing tasks raises several hardware-related and trust issues. Losing control over the servers and all data transfers within the cloud network calls for secure storage and computing mechanisms, as well as auditing techniques. Integrity-checking techniques have been around for long, but are not adequate for tackling cloud storage. Xiao and Xiao [300] overviewed Provable Data Possession (PDP) and Proofs of Retrievability (PoR) approaches, and they concluded that these approaches can only be applied to static files, therefore not being applicable in cloud systems. However, the research community started working toward dynamic approaches, which now include scalable PDP and dynamic PDP, but neither of them offer a complete set of characteristics that embrace all cloud requirements, such as public verifiability.

In terms of processing, the innovative homomorphic encryption [92,305] enables processing encrypted data directly on outsourced servers, that is, without the need to decrypt it. Although the concept of homomorphic encryption has been around for some time, it was not really known whether or not it was possible to fully achieve it in practice, until Gentry [90] proposed a fully homomorphic scheme. The main drawback of the scheme is the computational overhead, according to Schneier [245]. Nevertheless, the International Business Machines (IBM) corporation has taken a step forth in optimizing homomorphic encryption by releasing a software package called `HELib` [78]. Another approach that seems to be gaining terrain among cryptographers is Elliptic Curve Cryptography (ECC) [45], which uses public-key cryptography. On an ECC system, a 256-bit public key should be comparable to a 3072 RSA public key, thus having the potential to reduce overhead burden. All these fields require further research. Nevertheless, issues related with unreliable computing would still affect such approaches.

5.1.3 The virtualization layer

To address the virtualization issues, there is effort on devising stronger VMM solutions. As Pearce et al. [210] pointed out, virtualization issues should be handled with care and forethought. A strong solution can address confidentiality, integrity, and availability, but failures on one of these is enough to trigger potentially disastrous results. VMMs are large and complex while having thousands of lines of code. They mediate the creation and deletion of VMs, provide VMs with virtual resources, isolate running components as best

as they can, define virtualized networking, and provide the necessary virtual devices to route virtual traffic. They are a middleware layer between host OSes and guest OSes, hence revealing a considerable attack surface. There are four main research areas related with VMMs security [269], with one of them being now unsuitable, which will not be discussed herein.

The first is VMM minimizing, which consists in reducing the amount of code in the attempt of eliminating bugs and vulnerabilities. McCune et al. [164] built an hypervisor prototype with the main goal of executing self-contained security-sensitive code blocks. Moreover, Hua and Sakurai [115] also devised a lightweight hypervisor that isolates all Linux kernel modules into different memory address spaces. Tests showed that the solution outputs acceptable overhead.

The second main area is VMM hardening with additional code. With respect to VMM hardening, Liu et al. [153] presented a method of building a bridge firewall based on `iptables` to the Xen VMM. The method showed some performance obstacles. vShield from VMware, an industry product, puts itself in between VMs and virtual switches, therein inspecting all packets leaving the guest OSes. This approach scales up well even when SVAs are added on-the-fly as required [26].

Finally, in the latest VMM research area consists in giving VMs more direct access to hardware. Szefer et al. [269] proposed NoHype, a system that discards the dependency on VMM while maintaining VM concurrency with more contact with underlying hardware. At first, NoHype boots up VMs and provides necessary resources, but it then disengages them to run independently. This approach diminishes the virtual attack surface and, thus, provides enhanced security.

Regarding the particular issue of random number generation in virtualized environments, Kirkland [141] presented, in his talk, two main areas for improvement: PRNGs initial seeding and ongoing entropy gathering. With respect to PRNG seeding, a VMM could provide strong random initial seeds while booting VMs, but that might end up with seeds correlated with others on co-resident instances fired up within the same time-frame. Nevertheless, Moser [176] quickly introduced such a solution for OpenStack through its metadata service. A user could also provide its own seed, or it could be assembled by an external protocol. Concerning entropy gathering during the lifecycle of VMs, the most straightforward solution is to inject more entropy into entropy pools through alternative daemons or third-party protocols. Examples include the Entropy Gathering Daemon (EGD) and the HArdware Volatile Entropy Gathering and Expansion (HAVEGE), and the Entropy Broker and the Asynchronous Network Exchanged Randomness Daemon (aNerd), respectively. However, the latter protocols might suffer from network-based attacks, such as MitM and DoS.

There are more promising solutions for overcoming entropy starvation. The first solution is VirtIO RNG driver for KVM-based VMs. VirtIO is a feature of QEMU, an open-source emulator and virtualizer, and can be wired up to any entropy source on the host side. Although in public clouds this solution might not seem trustworthy enough under certain conditions, for private clouds it can be easily deployed using any type of entropy gathering solution. Just like Linux random devices, the driver can be exposed through the device `/dev/hwrng`. The second trendy and encouraging solution comes on a microprocessor chip of Intel, the microprocessor manufacturer, and can be used to solve random number generation issues. The solution, previously known as Bull Mountain but now code-named Intel Secure Key [120], is based on digital circuitry, a conditioner, and a cryptographically secure PRNG [273]. This Digital Random Number Generator (DRNG) takes thermal noise to output a raw stream of random bits at three gigabits per second, which is then remastered by the conditioner to improve randomness strength. The PRNG takes as seeds the outcome of the conditioner to produce 128 bit secure random numbers, which are attainable through the CPU `RdRand` instruction. Benchmarking results [139] seem to point highly scalable provisioning and quick throughput generation, while `dieharder` tests indicate good randomness quality.

5.1.4 The malware trends

Albeit mobility is a certain future for enterprise and cloud connectivity, and the fact that Android malware grew 2.577 % in 2012, mobile malware only takes a 0.42 % slice out of the top Web malware threats for 2012 [56]. Nevertheless, adequate attention should be given to each propagation medium. Malware writers are focusing on evasion techniques rather than finding ways into internal systems, because that is almost taken for granted, probabilistically speaking. Strength is being put on the Return On Investment (ROI). Thus, malware camouflage behavior might pave the path for next generation malware, and this definitely concerns virtualized environments as malware can change behavior on-the-fly if it detects such a presence, and the Trojan.Maljava is proof of it [134].

The way forward is uncertain, but one thing holds true: new strategies must be devised. O'Kane et al. [191] suggested incorporating behavioral information by focusing on what the suspected malware is doing rather than how it is doing it. Tracing behavior minimizes reliance on underlying technology and, thus, detection efficiency is not undermined and might therefore yield optimal analysis. Oyama et al. [202] have proposed a method incorporated into a thin hypervisor, but it is based on signatures. In agreement with O'Kane, effort has been put on tracing malware behavior. Vaccination tools have been developed. Authorship of Leder and Werner [144], the nonficker tool was developed to fully wipe out Conficker

from memory. Sun et al. [264] proposed a solution to detect anti-VM techniques based on the malware behavior. Zabidi et al. [307] provided a modular tool with anti-VM detector. Anecdotally, anti-sandbox, anti-VM, or anti-debug code can be backfired at by deploying tools capable of mimicking such systems in normal systems with the purpose of deterring malware, taking the concept to its extreme by trying to elude malware into thinking that underlying systems should be avoided rather to be infected. In this line of work, tools and methods have already been proposed [46,243,244]. The previous discussion illustrates the counterattack effort to mitigate malware propagation to the virtualization layer.

5.1.5 The Web-based access

In terms of Web-based technologies, there is a wide attack vector associated with the techniques used to deliver applications over the Internet. For a start, Web pages deliver mechanisms to outcome the flaws of underlying standards and protocols, such as HTTP statelessness. Not only that, but programmers usually oversee Web-related security measures in exchange for more fancy functionality. In such case, input validation is many times not correctly implemented, leaving behind holes for injecting SQL or JavaScript code. In addition, XSS and URL-guessing attacks explore the fragile `GET` method. Injection remains the main issue related with Web applications, but XSS is a growing trend in general. Nonetheless, even `POST` can be subverted by manipulating HTML hidden fields or stealing cookies. On top of those, HTTP should always be used over TLS, despite the existing attacks on the protocol. To mitigate Web applications vulnerabilities, Martin [161] suggested fostering software development teams with adequate security training. A SDLC must follow a solid approach by integrating security controls directly into the SaaS application stack. For instance, Data Loss Prevention (DLP) should be best deployed natively in the software. Moreover, intelligent logs must also be deployed, in order to then correlate them in a better way, ultimately resulting in a better and more focused perspective of a network health.

Dacosta et al. [68] proposed a one-time cookie stateless method to prevent session hijacking attacks. The method focuses on providing session integrity, but does not provide for data confidentiality and integrity. For those two, the one-time cookie method should be complemented with HTTPS. One-time cookie generates a unique token per request based on session keys, which are tied together using Hash-based Message Authentication Codes (HMACs). Furthermore, it borrows the concept of tickets from Kerberos, thus implementing symmetric cryptography and, in turn, requiring clients to keep encryption keys, which are saved on browsers. The approach main threat is, therefore, browser malware. The authors stated that previous mechanisms fail to address the requirement of highly distributed systems, thus putting the one-time cookie method on highlight for cloud systems with tests showing little overhead when compared to traditional insecure cookie approaches.

5.1.6 The network perimeter openness and dynamics

Cloud computing changes the networking perimeter and the underlying network security devices. Cloud computing is synonym of literally moving almost everything into the cloud, including applications for internal purposes or for enterprise customers, and data. To make all this available, a wide range of distinct types of connectivity is put in place. Not only that, but both the cloud and enterprise networks become lively dynamic with a plethora of devices generating traffic.

Shin and Gu [249] proposed CloudWatcher, a solution to overcome the issues posed by the diversity, complexity and dynamics of cloud networks. The solution is based on OpenFlow[2] and comprehends network traffic analysis techniques based on standard security controls. Moreover, Azmandian et al. [21] proposed an interesting and lightweight VMM-level IDS based on anomaly detection. The proposal uses the low-level architectural information visible to the VMM. It collects data from all VMs and then utilizes data mining techniques to classify traffic. Results showed an average accuracy of 93 % with 3 % of false alarms. Regarding security event management, the Open-Source Security Information Management (OSSIM) version of AlienVault is freely available on the Amazon EC2 marketplace [8]. It can be easily instantiated as it is provided through Amazon Machine Images (AMIs), which are specific image files of the Amazon cloud.

In order to reestablish trust in a mobile and cloud-enabled enterprise network, Amoroso [13] suggested a resource-centric model by adding idM, distributing resources across multiple clouds, and adding cloud assents along with network-based security controls. In the same context, Li and Clark [151] stated that device-based IDSes, application sandboxing and bare metal hypervisors, ontology firewalls, behavior-based detection and protection through VPN technology is not enough. Solutions of this scale have been proposed, but render incomplete approaches that take the cloud as a whole. They suggested tackling the problem with an Infrastructure-Centric Security Ecosystem with Cloud Defense (ICSECD). The ICSECD combines endpoint protection (including mobile devices) with cloud-based solutions and would be in charge of the enterprise. Components of the solution include application proxies, secure Web gateways, DLP engines, anti-malware engines, and cloud-based services that would interconnect the components in an intel-

[2] OpenFlow is an innovative routing technology that separates the data plane from the forwarding plane and is an enabler toward Software-Defined Networking (SDN).

ligence collaborative environment. Perhaps more interesting, Salah et al. [240] also provided an innovative solution that consists in deploying a security overlay network based on cloud computing. The security overlay network would contain security appliances and mechanisms, namely anti-*,[3] DDoS prevention and protection, IDSes and IPSes, and filtering spread across proxy servers and specialized appliances. A frontend security center would provide the tools to manage those assets, including a SIEM infrastructure, security policies, and an SSO proxy. Other protection measures could be easily provisioned due to the native cloud elasticity. A scalable load balancer is put on the network input point, while output traffic is forwarded to customers—acting like a big proxy. The network of the customer restricts incoming traffic to only allow traffic coming from the security overlay network. Endpoint protection is maintained within the customer network, while managing and monitoring internal network health as well as normal production servers. Their real tests depict advantages that include network concealment against reconnaissance techniques, detection and prevention effectiveness, flexibility for additional resources and higher performance, and costs reduction. The security overlay network design assumes a secure cloud environment. This is the main drawback of the solution, as this article has already demonstrated.

5.1.7 Balancing auditability with trust

As extensively discussed in this article, trust is a barrier that transversely extends throughout the whole cloud components and stakeholders. Chen et al. [47] invoked the term mutual auditability to refer to collaborative monitoring with the purpose of proving reciprocal trustworthiness. In other words, rather than focusing the auditability in the customer-provider direction, a bidirectional approach is adopted. This can improve incident response and recovery times, since both providers and customers can be the source or target of an attack. Moreover, Rasmusson and Aslam [220] have provided a novel solution that uses Trusted Platform Module (TPM) technology to prevent a provider from eavesdropping VMs. In addition, it also allows the provider to conveniently monitor malicious behavior of VMs through a set of probes that are agreed between customers and providers. Such agreement is conducted with an initial negotiation protocol. Each probe is inlined to the customer VM code with a binary code inliner while maintaining due separation of the protected memory blocks of each one, thus preserving the privacy of customers. Those probes can be installed for any purpose, like checking for network attacks or licensed code, or probing for malware, or for providing useful audit information for both sides. Therefore, the solution contemplates two perspective: it protects the provider from the customer but also the other way around. Thus, beyond the one-way integrity-checking methods being useful, for the customer side that is, both parties on the cloud agreement are required to be satisfied. This aspect needs to be emphasized in future audit methods.

[3] Anti-* stands for anti-spam, anti-virus, anti-spyware and anti-phishing.

5.1.8 The privacy state

The current privacy state is not yet well understood. As Pearson [211] pointed out, there is an ongoing change in privacy and it is the biggest since the eighties. Efforts are being put in fairness, accountability and increased protection by policy makers [211]. However, CISPA and previous rejected acts say otherwise. There is an increasingly government desire to mass supervise data from Internet users in the scope of cyberthreat protection. Although the ultimate goal is to actively prevent or mitigate cyberthreats, such as APTs, the privacy perimeter of each individual and entity is breached in such case, and cloud environments do not escape such attempts. Moreover, current transborder data flow restrictions, geographic location of data storage and computing, and data under law enforcement perspective adds more uncertainty to this matter. In addition, VM-level security holes and deficient sanitization are also included in the current unstable privacy state of cloud computing.

5.1.9 Standards and open-source projects

The rapid adoption of cloud computing resulted in a many cloud proprietary formats developments, in turn giving out the fear of vendor lock-in. The need to standardize formats in clouds is clear. To that end, leaders around the world started to work on various open standards. The Open Data Center Alliance (ODCA), founded in 2010, aims to speed up the migration of current cloud environments to interoperable and standardized cloud systems, and the Open Cloud Initiative (OCI) [193] aims to legally regulate such standards.

The OASIS created SAML [241], which defines an open data format for exchanging authentication- and authorization-related information. It adopts the concept of IdP, which provides an identity assertion on behalf of an entity to a service provider. The service provider then makes an access control decision based on such assertion, allowing, or not, an entity to access a service. VMware and other players of the virtualization field created the Open Virtualization Format (OVF) [288]. It is a platform-independent open format for packaging and distributing VMs, with basis on efficiency, extensibility and security characteristics.

A management interface standard named Cloud Data Management Interface (CDMI) [253] was created by the Storage Networking Industry Association (SNIA). It defines

a frontend interface for cloud administrators that allows managing containers, accounts, security accesses, and information with respect to monitoring and billing. It allows to create, retrieve, update and delete data components (including metadata) from clouds. In addition, cloud customers can use the interface to manage data containers and the data contained in them, therein discovering the capabilities and offerings of a service. The Open Grid Forum created the Open Cloud Computing Interface (OCCI) [189]. In broad terms, OCCI is a protocol and API for all kinds of management tasks. It focuses on integration, portability, interoperability, and innovation, while maintaining a wide opening for extensibility. Moreover, it is suitable to serve many cloud service delivery models, including IaaS, PaaS, and SaaS.

There are various open-source projects available on the market. Of note are the following. The first is the open-source project named OpenNebula [194], released in 2008, which aims at providing a one-size-fits-all solution for virtualized data center infrastructures and enterprise private clouds. It provides a comprehensive management layer to automate and orchestrate networking, storage, virtualization, monitoring and user management. On the same track, founded by Rackspace Hosting and the National Aeronautics and Space Administration (NASA) in 2010, OpenStack [195] aims to deliver solutions for all types of clouds on a massively scalable open-source cloud operating system. It controls large pools of compute, storage, and networking resources through a dashboard, while empowering cloud users with a Web interface access. Developers can build their own tools to manage their resources using the OpenStack API or the Amazon EC2 compatibility API. Finally, the CloudStack [15] project—maintained by the Apache—is designed for IaaS private or hybrid clouds, aiming at deploying and managing large networks of VMs. It is a turnkey solution that supports popular virtualization solutions, such as VMware, KVM, and Xen. This project is alive since 2012.

5.1.10 Final remarks

The Internet bears a great number of security threats. The Spamhaus case described earlier was solved with anycast, leveraging the cloud elasticity as a countermeasure as well. But clouds are elastic to some extent as the race in power foretells a challenge for next generation attacks like the Spamhaus DDoS. The multi-billion dollar online crime industry [267] supports an increasingly sophisticated black market that sells powerful crime packs. Because vulnerabilities are quickly included in those packs, it is utterly important for cloud providers to ensure rapid deployment of security patches for their managed infrastructures. Cloud environments entail a brand new class of threats that are magnified when compared to other similar systems and include on top the aforementioned Internet threats. For instance, due to such a blur security state, it is yet a high risk for the financial industry to move onto cloud environments for their highly sensitive businesses [170]. Therefore, by taking into account the previous discussions, one can say that research on the field will continue to tackle the security issues toward the goal of more secure, reliable, and trustworthy cloud environments.

5.2 Recommendations for practitioners

A few practical recommendations are next handed out for cloud providers, cloud customers and cloud users. Without the human factor, malicious actors would have to resort to more sophisticated ways to get into a protected network. For example, if spear-phishing would not be successful, then the remaining attack vectors would consist in exploring publicly accessed infrastructures and applications or more extreme physical break-ins. Moreover, there should always be benevolence when browsing Internet sites for their malware threats, what kind of contents are shared therein and which credentials are chosen for applications. Several distinct characters should be used to either construct a string with enough entropy or a logical phrase that can be easily remembered. These two approaches provide strong passwords to use in Web sites or enterprise applications, which should be properly instructed to employees. In addition, applications should not allow the use of common passwords, enforcing strong passwords therein. As pointed out by Goodin [95], increasing the password length character by character exponentially increases cracking time, eventually hitting the so-called exponential wall of brute-force cracking. The main design goal of the Secure Hash Algorithm-1 (SHA-1) and Message Digest 5 (MD5) is to be plain fast while using minimal computing resources. This eases brute-force attacks and, therefore, single iteration cryptographic hash functions are just not enough to save salted and hashed passwords. Instead, slower, multi-iteration hashing algorithms should be used, like `bcrypt` [105]. Such an approach can dramatically improve defense against brute-force password cracking techniques in enterprise or cloud cryptosystems, but then computational overhead would also increase. Therefore, the trade-off between performance and security level should be wisely considered.

Regarding the BYOD paradigm, Cisco [56] stated that employees devices should be analyzed by their employer, assuring that those are not rooted or jailbreaked. This can avoid malware propagation by restricting users to install trustworthy applications from official distribution stores. Any programmer can be its own publisher and, therefore, anyone can write pieces of malicious code. To avoid this, official channels ensure applications integrity and check for malicious code before releasing applications to the market. Such policy enforcement can help to reduce SaaS applications and network perimeter risks. Still regarding the net-

work perimeter, Anstee [14] said that one common response against DDoS is the belief that firewall and IPS appliances protect against such threat. Unfortunately, this is not true. Instead, a layered approach with DDoS mitigation solutions should be deployed outside of those security devices. These solutions should maintain minimal state in order not to consume resources when under attack, in contrast with stateful methods of firewalls not suitable for the job. The concepts enlightened by Li and Clark [151] and Salah et al. [240] seem to align with this network arrangement, foretelling developments in this regard. Contacting a specialized cloud provider with DDoS mitigation services is optimal, such as CloudFlare, because their infrastructures have sufficient capacity for absorbing the attacks impact.

In order to address the issue of proprietary formats, it is recommended for cloud providers to support and adhere to open cloud standards by making solutions compatible with each other. Although open-sourcing is risky for its open code that can ease reverse engineering, in the case of cloud computing it seems to be a good choice. Besides being cheaper, it would contribute to an interoperable and standardized wide cloud ecosystem throughout cloud providers, henceforth overcoming the proprietary lock-in. In addition, it could propel the security community to focus on single standards rather than trying to address each vendor issues. The OpenStack [195] project community think that an open development model is the only way to foster such an ecosystem. In fact, it would also give way to integrate private clouds with public, creating a proper foundation for hybrid clouds to thrive [56]. Another recommendation is related with the resource recycling. IP addresses and physical (e.g., hard disks) or virtual resources (e.g., VMs) should not be handed out to new customers while there are remnants of previous usage, such as request load, data, or configurations [42]. This can inadvertently leak information, therein breaching privacy of impacted customers.

Other recommendation is to adopt and implement Two-Factor Authentication (2FA). In fact, big players such as Google [101], Facebook and Apple, motivated by the weak password choices and security intrusions, have already deployed it in addition to basic username and password authentication. 2FA builds upon the premise of "*something you know*", the username and password, with "*something you have*", a physical token. The physical token refreshes an access code periodically with basis on a time-based algorithm, producing the so-called Time-Based One-Time Passwords (TOTPs). The authentication server also runs the same algorithm with the same initial pre-shared key so as to generate synchronized codes with the token. The code is asked after verifying the username and password combination. RSA SecurID [235] is an example of a physical token whose sole purpose is to generate TOTPs. The 2FA would be suitable given the current BYOD paradigm, because smartphones are able to produce such codes via an application or receive them via SMS.

A careful assessment between all available cloud deployment models must be considered in order to balance factors weights, including advantages and disadvantages, with focus on the security perspective. For that, trusted third-party auditors are recommended. CIOs should close enterprise open DNS resolvers so as to avoid participating in DNS reflection and amplification attacks and should consider security as a forefront priority. Security should be deployed as a transverse aspect throughout both hardware and software, and not as an appendix. Security should be considered to be part of a full SDLC approach [161,165] and span across all software engineering phases.

Despite all security measures available nowadays to counterattack the several issues, one should always have in mind the following important truth: no system is 100 % secure. A system is as secure as its weakest link. Past security events have proven that, no matter what kind of new technology is invented, the truth is that technology may be flawed due to human error. Malicious actors have limitless creativity in devising workaround alternatives to attain their objectives. Thus, the possibility of an unknown threat that may be exploited via an unknown attack vector is alarmingly present—zero-day vulnerabilities are hard to detect. In fact, the CSA defines unknown risk profile as one of the top threats to cloud computing [62], but this spans to other computer systems as well. Many companies might overlook security issues if the short-term benefits outweigh the risks taken. In this scenario, unknown risks arise when security matters are not prioritized or are put in hold. The CSA suggests that the information about who is sharing a cloud infrastructure is important to assess security risks and should be complemented with security logs. According to the alliance, the potential of unknown threats is larger in cloud environments and that alone should provide enough motivation for considering security as one of the top priorities for cloud providers [306].

5.3 Ideally secured cloud environments

With basis on the study presented in this article, a straightforward and conceptual outline of how cloud environments should be secured is discussed in this subsection. For a start, cloud providers must ensure that all data stored in the cloud is not spied on by governmental agencies. The leaked controversial PRISM program of the NSA placed some pressure on some big IT players, namely Microsoft and Apple, who supposedly gave access to private data belonging to users. For some, the trust deposited on the providers has diminished, not to mention the potentially legal violations. To avoid this, international laws should be put forward so as to address such cases and ensure user privacy in a lawful manner. In addition, customers are not guaranteed to get feedback from

providers in situations of subpoenas or urgent matters. In any case, everything in the cloud should be encrypted, with the encryption keys in charge of the customers. The most secure scenario is to use hybrid clouds, so as to save sensitive information on-premises, like AD or LDAP credentials. CloudStack permits mapping CloudStack accounts to the corresponding LDAP accounts. In addition, an off-premises cloud infrastructure could act like the inner enterprise gateway while offering proxy S_{ec}aaS solutions in a hybrid model. As an alternative, a trusted third-party could be in charge of those tasks and of others related with auditing, to ensure a continuous untroubled service subscription. In this third-party regard and because the number of IdPs is growing at a fast pace, both Internet users and enterprises should go for IdP-based authentication. This avoids having to manage multiple credentials for multiple SaaS applications and can allow enterprises to use their internal domain users to login on those applications. The McAfee Cloud Single Sign-On product can achieve that while enforcing corporate standards.

An ideal interoperable and flexible cloud ecosystem would require data migrations to be the responsibility of the cloud providers, but encryption keys would remain with the customer or a trusted third-party as aforementioned. Such mode of operation is already backed up by the encryption scheme that MEGA offers, on which persistent storage on the cloud is encrypted. Moreover, key generation is done during user registration, but with a twist. Entropy is collected from mouse movements and keystroke timings which, despite comprising a small entropic input set, comes from the user side. This entropy is a crucial ingredient for generating cryptographically strong random numbers, in turn strengthening cryptographic keys, which should be programatically generated and stored on the client side and never on the server side. In contrast, Amazon EC2 generates keys on the cloud, and it is not clear whether or not they are eliminated afterward. OpenStack does the same. All this is irrelevant if users passwords are not correctly stored by means of slow hashing algorithms like `bcrypt` using appropriately large salts. Furthermore, data standards would ensure easy data transfers between clouds and seamless integration with management interfaces, SaaS applications or PaaS APIs or IDEs. For remote computational tasks, homomorphic schemes seem to be in the right direction, though still far from a practical implementation. To achieve a securer cloud, the trade-off between performance and security should be skewed for the latter. For more stringent computational tasks, an on-premises data center could be the right option for high-performance computing clouds to cope with the overhead, like the Nebula One solution.

Regarding virtualization, the need for more secure hypervisors is clear. Some research works discussed in the previous section offer good pointers, but such methods and techniques presented therein must be adopted by the big virtualization players. Not only that, but hardware vendors should also support virtualization technologies. This is the case of hardware-assisted virtualization of Intel and AMD with their VT-x and AMD-V CPU technologies, respectively, which dismiss the use of binary translation. Virtualization software supported by hardware with the same goal could completely isolate VMs and prevent cross-VM attacks. In addition, more security controls could be added to VMs and outsourced networks. Amazon EC2 offers a firewall and CloudWatch, which monitors CPU and disk usage as well as network activity per VM instance. For more demanding cases, some set of clustered physical servers can be allocated to particular customers to house an entire virtual data center. Such a cluster would be segregated from the remaining part of the cloud, therefore providing higher security. This is the case of Amazon VPC. From the customer point of view, CSIRTs would require to conveniently monitor outsourced infrastructures. Networking equipments, such as routers and switches, would have to integrate secure mechanisms for extending the internal network perimeter to provider-hosted clouds. Ultimately, the goal of achieving a functioning and secure hybrid cloud would be easier if security was deployed within the networking fabric. The Cisco Cloud Services Router 1000V series [57] is the prime example to lower the adoption barrier of the hybrid cloud deployment model.

In terms of access to clouds, management interfaces and remote access protocols are currently used. The Plesk Panel interface, for instance, is used for pumping up hosted sites, for which many are on cloud systems. When it was recently found vulnerable, a botnet exploiting the vulnerability was shutdown [223]. It is therefore imperative to patch vulnerabilities as quickly as possible. For clearing the mist on these cases, SLAs should include security aspects and cover unexpected situations. The expected average time to patch vulnerabilities should be included. With respect to Remote Desktop Protocol (RDP) and SSH access to VMs, the complexity of credentials or key management cannot be circumvented. As in Amazon EC2, private keys are sent via HTTPS to the client side. One is able to use the same key for every instance or create a new one. To diminish the chances of an adversary successfully bypassing authentication, 2FA should be deployed onto two places. The first is account authentication, as mentioned in the previous subsection. The second is within the kernel of operating systems. 2FA can be easily added to the Linux kernel Pluggable Authentication Module (PAM). On remote connections, such a second layer would definitely help prevent breaches.

For an utopic enterprise network, companies adopting the BYOD should enforce security policy on user devices while offering the means to protect (e.g., using anti-virus) the device and access backend cloud applications conveniently. In the worst or sensitive cases, like having smartphones unlocked or jailbreaked, enterprises should completely cut off their access. To make a cloud system interoperable with

nowadays devices, applications should be multi-platform, which include classic Windows, Mac OS and Linux operating systems, but also mobile systems, such as Android, iOS, and Symbian. In this case, mobile applications certainly lack the functionalities of more robust computer software. Consuming APIs through mobile applications should be done in a fail-safe manner while notifying the user of unexpected or abnormal events. These are *best-effort* approaches to cope with the current mobile trends.

In terms of the physical security of the perimeter, current data center security policies seem to be sufficient (they are perhaps the most matured policies). Nevertheless, in terms of the digital perimeter, the data center construction design for the network perimeter should take into account that the total aggregated bandwidth must be sufficient for inter-cluster communications and for both upstream and downstream data bursts. In addition, new security controls—including anti-virus, firewalls/IPSes and IDSes—not relying on stateful inspection for mitigating malware dynamics and flooding attacks would certainly help too.

Cloud providers should strive to meet body standards and quantify the risk of moving to the cloud, beyond advertising the features their clouds have. For this task, the NIST risk framework [187], which contemplates various steps to formally define a security risk management workflow, provides an appropriate baseline for the future in this regard. Microsoft also released the Cloud Security Readiness Tool (CSRT) for assessing what enterprises could expect if they adopted a cloud solution to replace their IT systems.

6 Conclusions

The hype of cloud computing paradigm is pumping the IT industry toward a long-envisioned era. Having it as the fifth utility, following water, electricity, gas and telephony grids, is being widely accepted throughout businesses. The commodity of delivering services on-demand is a practical solution for many low- to medium-sized enterprises, mainly lowering general infrastructure costs and augmenting business productivity. Nevertheless, as with any new technology, cloud environments are still subject to improvements, namely regarding security.

Cloud computing is nowadays dominated by a large number of challenges. Due to its rapid growth and because virtualization is a relatively new technology, a burst of security issues have been discovered and studied by both the academia and industry. There is a general preoccupation surrounding the adoption of cloud-related products. To accomplish the objective of delivering secure cloud environments, patching those security issues is a priority. In addition, cybercriminals follow trends, and cloud computing certainly does not escape that course. Cybercrime is increasingly becoming more sophisticated. Malicious actors team up and form malware assembly lines, on which each one has a specific task, like writing the malware, define spam tactics, design a social engineering component, and so on. The enterprise network security is currently under highly volatile conditions, and the security landscape gets darker when mixing up cloud environments with the rate of the increasing and improved cybercriminality.

In this article, the state-of-the-art on cloud security issues was discussed. A broad scope analysis of the literature was presented, which included studies from the academia and from the industry. Each study was reviewed to determine its aim and harvest the materials needed to better cover all topics in the security state of cloud environments from several perspectives. Basic concepts related with clouds were also explained so as to better provide the basis to understand this article. They were, nonetheless, and whenever possible, introduced with an especial focus on the security topic. Several real-life examples were included to provide rationale for the discussions and to illustrate the impact of the security issues.

The analysis of the literature bespeaks a clear interest toward addressing cloud security issues. A strong will and *momentum* to take a leap forth in devising secure clouds is extracted from the studies, revealing intentions from both the academia and the industry. As this field matures, it is expected to see more robust methods to cope with the stringent requirements of cloud environments. Although cloud computing is already a mainstream technology and it is yet growing, it is also expected to see it settle down, converging its current diversity into more streamlined solutions. This will enable a better understanding of the security state and will allow dissipating doubts on the technology. Until then, customers might not fully experience the cloud computing technology and cloud security issues must be resolved. History has proved that security should be a top priority and that the research and development on this area is partially motivated by issues faced along the way, which seems to apply in this case also.

Acknowledgments We would like to thank all the anonymous reviewers for constructively criticizing this work.

References

1. 57un Blog: A BIG Password Cracking Wordlist. https://57un.wordpress.com/2013/03/09/a-big-password-cracking-wordlist/. Accessed May 2013 (2013)
2. Aguiar, E., Zhang, Y., Blanton, M.: An Overview of Issues and Recent Developments in Cloud Computing and Storage Security, pp. 1–31. Springer, Berlin (2013)
3. Ahuja, S.P., Komathukattil, D.: A survey of the state of cloud security. Netw. Commun. Technol. **1**(2), 66–75 (2012). doi:10.5539/nct.v1n2p66
4. Aihkisalo, T., Paaso, T.: Latencies of service invocation and processing of the REST and SOAP web service interfaces. In:

IEEE 8th World Congress on Services (SERVICES), pp. 100–107. Honolulu, HI, USA (2012). doi:10.1109/SERVICES.2012.55
5. Al-Aqrabi, H., Liu, L., Xu, J., Hill, R., Antonopoulos, N., Zhan, Y.: Investigation of IT security and compliance challenges in security-as-a-service for cloud computing. In: 15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), pp. 124–129. Shenzhen, Guangdong, China (2012). doi:10.1109/ISORCW.2012.31
6. Alert Logic: State of Cloud Security Report: Targeted Attacks and Opportunistic Hacks. http://www.alertlogic.com/resources/security-intelligence-newsletter/download-cloud-security-report-spring2013/ (2013). Accessed Apr. 2013
7. AlFardan, N., Bernstein, D., Paterson, K., Poettering, B., Schuldt, J.: On the Security of RC4 in TLS. http://www.isg.rhul.ac.uk/tls/index.html (2013). Accessed Apr. 2013
8. AlienVault: OSSIM Website. https://aws.amazon.com/marketplace/pp/B00BIUQRGC/ (2013). Accessed May 2013
9. Amazon: Amazon Web Services: Overview of Security Processes. http://s3.amazonaws.com/aws_blog/AWS_Security_Whitepaper_2008_09.pdf (2011). White Paper. Accessed Sept. 2012
10. Amazon: Amazon Elastic Compute Cloud (Amazon EC2). https://aws.amazon.com/ec2/ (2012). Accessed Apr. 2013
11. Amazon: Amazon Virtual Private Cloud (Amazon VPC). http://aws.amazon.com/vpc/ (2012). Accessed Sept. 2012
12. Amazon Web Services Discussion Forums: Low Entropy on EC2 Instances— Problem for Anything Related to Security. https://forums.aws.amazon.com/thread.jspa?messageID=249079 (2011). Accessed Apr. 2013
13. Amoroso, E.: From the enterprise perimeter to a mobility-enabled secure cloud. IEEE Secur. Priv. **11**(1), 23–31 (2013). doi:10.1109/MSP.2013.8
14. Anstee, D.: Q1 Key Findings from ATLAS. http://www.arbornetworks.com/corporate/blog/4855-q1-key-findings-from-atlas (2013). Accessed Apr. 2013
15. Apache: CloudStack Website. https://cloudstack.apache.org/ (2013). Accessed May 2013
16. Apprenda: Apprenda Website. http://apprenda.com (2013). Accessed Apr. 2013
17. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010). doi:10.1145/1721654.1721672
18. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Zaharia, M.: Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28. Electrical Engineering and Computer Sciences University of California (2009)
19. Ateniese, G., Di Pietro, R., Mancini, L.V., Tsudik, G.: Scalable and efficient provable data possession. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Networks, pp. 9:1–9:10. ACM, New York, NY, USA (2008)
20. Aviram, A., Hu, S., Ford, B., Gummadi, R.: Determinating timing channels in compute clouds. In: Proceedings of the ACM Workshop on Cloud computing, Security, pp. 103–108 (2010). doi:10.1145/1866835.1866854
21. Azmandian, F., Moffie, M., Alshawabkeh, M., Dy, J., Aslam, J., Kaeli, D.: Virtual machine monitor-based lightweight intrusion detection. SIGOPS Oper. Syst. Rev. **45**(2), 38–53 (2011). doi:10.1145/2007183.2007189
22. Back, G., Hsieh, W.C.: The KaffeOS Java runtime system. ACM Trans. Program. Lang. Syst. **27**(4), 583–630 (2005). doi:10.1145/1075382.1075383
23. Backstrom, L., Dwork, C., Kleinberg, J.: Wherefore art thou R3579X?: anonymized social networks, hiddenpatterns, and structural steganography. In: Proceedings of the 16th International Conference on World Wide Web, pp. 181–190. ACM, New York, NY, USA (2007). doi:10.1145/1242572.1242598
24. Bahram, S., Jiang, X., Wang, Z., Grace, M., Li, J., Srinivasan, D., Rhee, J., Xu, D.: DKSM: subverting virtual machine introspection for fun and profit. In: 29th IEEE Symposium on Reliable Distributed Systems, pp. 82–91. IEEE Computer Society, Washington, DC, USA (2010). doi:10.1109/SRDS.2010.39
25. Banerjee, P., Friedrich, R., Bash, C., Goldsack, P., Huberman, B., Manley, J., Patel, C., Ranganathan, P., Veitch, A.: Everything as a service: powering the new information economy. Computer **44**(3), 36–43 (2011). doi:10.1109/MC.2011.67
26. Basak, D., Toshniwal, R., Maskalik, S., Sequeira, A.: Virtualizing networking and security in the cloud. SIGOPS Oper. Syst. Rev. **44**(4), 86–94 (2010). doi:10.1145/1899928.1899939
27. Begum, S., Khan, M.: Potential of cloud computing architecture. In: International Conference on Information and Communication Technologies, pp. 1–5. IEEE (2011). doi:10.1109/ICICT.2011.5983572
28. Behl, A.: Emerging security challenges in cloud computing: an insight to cloudsecurity challenges and their mitigation. In: World Congress on Information and Communication Technologies, pp. 217–222. IEEE (2011). doi:10.1109/WICT.2011.6141247
29. Behl, A., Behl, K.: Security paradigms for cloud computing. In: 4th International Conference on Computational Intelligence, Communication Systems and Networks, pp. 200–205. IEEE (2012). doi:10.1109/CICSyN.2012.45
30. Belqasmi, F., Singh, J., Glitho, R.: SOAP-based vs. RESTful web services: a case study for multimedia. IEEE Internet Comput. **16**(4), 54–63 (2012). doi:10.1109/MIC.2012.62
31. Bentounsi, M., Benbernou, S., Atallah, M.: Privacy-preserving business process outsourcing. In: IEEE 19th International Conference on Web Services, pp. 662–663. IEEE (2012). doi:10.1109/ICWS.2012.34
32. Bernstein, D., Vij, D.: Intercloud security considerations. In: IEEE 2nd International Conference on Cloud Computing Technology and Science, pp. 537–544. IEEE Computer Society, Washington, DC, USA (2010)
33. Bin Mat Nor, F., Jalil, K., Manan, J.L.: An enhanced remote authentication scheme to mitigate man-in-the-browser attacks. In: International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), pp. 271–276. Kuala Lumpur, Malaysia (2012). doi:10.1109/CyberSec.2012.6246086
34. Boampong, P.A., Wahsheh, L.A.: Different facets of security in the cloud. In: Proceedings of the 15th Communications and Networking Simulation Symposium, pp. 5:1–5:7. Society for Computer Simulation International, San Diego, CA, USA (2012)
35. Bowers, K.D., Juels, A., Oprea, A.: HAIL: a high-availability and integrity layer for cloud storage. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 187–198. ACM, New York, NY, USA (2009). doi:10.1145/1653662.1653686
36. Box: Box Website. https://www.box.com/ (2013). Accessed Apr. 2013
37. Bradbury, D.: Shadows in the cloud: Chinese involvement in advanced persistent threats. Netw. Secur. **2010**(5), 16–19 (2010). doi:10.1016/S1353-4858(10)70058-1
38. Brito, H.: Pentagon Creating "Rules of Engagement" for Responding to Advanced Attackers. Mandiant M-Unition (2013)
39. Bugiel, S., Nürnberger, S., Pöppelmann, T., Sadeghi, A.R., Schneider, T.: AmazonIA: when elasticity snaps back. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, pp. 389–400. ACM, New York, NY, USA (2011). doi:10.1145/2046707.2046753

40. Carriço, P.: Low entropy on VMs. . . http://blog.pedrocarrico.net/post/17026199379/low-entropy-on-vms (2012). Accessed May 2013
41. Carroll, M., Kotzé, P., Van der Merwe, A. (2011). Secure virtualization—benefits, risks and controls. In: Leymann, F., Ivanov, I., van Sinderen, M., Shishkov, B. (eds.) CLOSER, pp. 15–23. SciTePress
42. Casale, A.: The Dangers of Recycling in the Cloud. TheMakegood (2013)
43. Chen, C.C., Yuan, L., Greenberg, A., Chuah, C.N., Mohapatra, P.: Routing-as-a-Service (RaaS): a framework for tenant-directed route control in data center. In: Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM), pp. 1386–1394 (2011) doi:10.1109/INFCOM.2011.5934924
44. Chen, D., Zhao, H.: Data security and privacy protection issues in cloud computing. In: International Conference on Computer Science and Electronics Engineering, vol. 1, pp. 647–651. IEEE (2012). doi:10.1109/ICCSEE.2012.193
45. Chen, T.H., lien Yeh, H., Shih, W.K.: An advanced ECC dynamic ID-based remote mutual authentication scheme for cloud computing. In: 5th FTRA International Conference on Multimedia and Ubiquitous Engineering (MUE), pp. 155–159. Crete, Greece (2011). doi:10.1109/MUE.2011.69
46. Chen, X., Andersen, J., Mao, Z., Bailey, M., Nazario, J.: Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware. In: IEEE International Conference on Dependable Systems and Networks (DNS) With FCTS and DCC, pp. 177–186. Anchorage, AK, USA (2008). doi:10.1109/DSN.2008.4630086
47. Chen, Y., Paxson, V., Katz, R.H.: What's New About Cloud Computing Security? Technical Report UCB/EECS-2010-5. EECS Department, University of California, Berkeley (2010). http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-5.html
48. Chonka, A., Xiang, Y., Zhou, W., Bonti, A.: Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks. J. Netw. Comput. Appli. **34**(4), 1097–1107 (2011). doi:10.1016/j.jnca.2010.06.004
49. Choudhary, V.: Software as a service: implications for investment in software development. In: 40th Annual Hawaii International Conference on System Sciences, p. 209a. IEEE Computer Society, Washington, DC, USA (2007). doi:10.1109/HICSS.2007.493
50. Chow, R., Golle, P., Jakobsson, M., Shi, E., Staddon, J., Masuoka, R., Molina, J.: Controlling data in the cloud: outsourcing computation without outsourcing control. In: Proceedings of the ACM Workshop on Cloud Computing Security, pp. 85–90. ACM, New York, NY, USA (2009). doi:10.1145/1655008.1655020
51. Christodorescu, M., Sailer, R., Schales, D.L., Sgandurra, D., Zamboni, D.: Cloud security is not (just) virtualization security: a short paper. In: Proceedings of the ACM Workshop on Cloud Computing Security (CCSW), pp. 97–102. ACM, Chicago, IL, USA (2009). doi:10.1145/1655008.1655022
52. Chung, H., Park, J., Lee, S., Kang, C.: Digital forensic investigation of cloud storage services. Digit. Investig. (2012). doi:10.1016/j.diin.2012.05.015. Available online on 23 Jun. 2012
53. Cisco: Cisco Data Center Infrastructure 2.5 Design Guide. http://www.cisco.com/univercd/cc/td/doc/solution/dcidg21.pdf (2007). Accessed Oct. 2012
54. Cisco: Data Center Power and Cooling. http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns944/white_paper_c11-680202.pdf (2011). White Paper. Accessed Sept. 2012
55. Cisco: Cisco Global Cloud Index: Forecast and Methodology, 2011–2016. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns1175/Cloud_Index_White_Paper.pdf (2012). White Paper. Accessed Apr. 2013
56. Cisco: 2013 Cisco Annual Security Report. http://www.cisco.com/en/US/prod/vpndevc/annual_security_report.html (2013). Accessed Apr. 2013
57. Cisco: Cisco Cloud Services Router 1000V Series. http://www.cisco.com/en/US/products/ps12559/index.html (2013). Accessed Jul. 2013
58. Citrix: Citrix Website. https://www.citrix.com/products.html?ntref=hp_nav_us (2013). Accessed Jun. 2013
59. CloudBees: CloudBees Website. http://www.cloudbees.com/ (2013). Accessed Apr. 2013
60. Corbató, F.J., Vyssotsky, V.A.: Introduction and overview of the Multics system. In: Proceedings of the Fall Joint Computer Conference, pp. 185–196. ACM, New York, NY, USA (1965)
61. Coronado, C.: Blackhole Exploit Kit Leverages Margaret Thatcher's Death. Trend Micro (2013)
62. CSA: Top Threats to Cloud Computing. https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf (2010). Accessed Sept. 2012
63. CSA: Security Guidance for Critical Areas of Focus in Cloud Computing v3.0. https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf (2011). Accessed Sept. 2012
64. CSA: The Notorious Nine Cloud Computing Top Threats in 2013. https://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf (2013). Accessed Jul. 2013
65. Cuckoo Website: Cuckoo. http://www.cuckoosandbox.org/ (2013). Accessed Apr. 2013
66. Curran, K., Dougan, T.: Man in the browser attacks. Int. J. Ambient Comput. Intell. **4**(1), 29–39 (2012). doi:10.4018/jaci.2012010103
67. Czajkowski, G., Daynàs, L.: Multitasking without compromise: a virtual machine evolution. ACM SIGPLAN Not. **47**(4a), 60–73 (2012). doi:10.1145/2442776.2442785
68. Dacosta, I., Chakradeo, S., Ahamad, M., Traynor, P.: One-time cookies: preventing session hijacking attacks with stateless authentication tokens. ACM Trans. Internet Technol. **12**(1), 1:1–1:24 (2012). doi:10.1145/2220352.2220353
69. Dahbur, K., Mohammad, B., Tarakji, A.B.: A survey of risks, threats and vulnerabilities in cloud computing. In: Proceedings of the International Conference on Intelligent Semantic Web-Services and Applications, pp. 12:1–12:6. ACM, New York, NY, USA (2011)
70. Darrow, B., Higginbothamm, S.: What We'll See in 2013 in Cloud Computing. GigaOM (2012)
71. de Borja, F.: Nebula One Seeks To Reinvent Cloud Computing. CloudTimes (2013)
72. Dhage, S.N., Meshram, B.B., Rawat, R., Padawe, S., Paingaokar, M., Misra, A.: Intrusion detection system in cloud computing environment. In: Proceedings of the International Conference & Workshop on Emerging Trends in Technology, pp. 235–239. ACM, New York, NY, USA (2011). doi:10.1145/1980022.1980076
73. Dinesha, H., Agrawal, V.: Multi-level authentication technique for accessing cloud services. In: International Conference on Computing, Communication and Applications, pp. 1–4. IEEE (2012). doi:10.1109/ICCCA.2012.6179130
74. Ding, X., Zhang, L., Wan, Z., Gu, M.: De-anonymizing dynamic social networks. In: IEEE Global Telecommunications Conference, pp. 1–6. IEEE (2011). doi:10.1109/GLOCOM.2011.6133607
75. Doel, K.: Scary Logins: Worst Passwords of 2012 and How to Fix Them. SplashData (2012)
76. Dong, T.: Android. Dropdialer. https://www.symantec.com/security_response/writeup.jsp?docid=2012-070909--0726-99 (2012). Accessed Apr. 2013

77. Doroodchi, M., Iranmehr, A., Pouriyeh, S.: An investigation on integrating XML-based security into Web services. In: 5th IEEE GCC Conference Exhibition, pp. 1–5. IEEE (2009)
78. Ducklin, P.: HElib. SOPHOS Nakedsecurity (2013)
79. Duncan, A., Creese, S., Goldsmith, M.: Insider attacks in cloud computing. In: IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 857–862. IEEE Computer Society, Washington, DC, USA (2012). doi:10.1109/TrustCom.2012.188
80. Dykstra, J., Sherman, A.T.: Acquiring forensic evidence from infrastructure-as-a-service cloud computing: exploring and evaluating tools, trust, and techniques. Digit. Investig. **9, Supplement**(0), S90–S98 (2012). doi:10.1016/j.diin.2012.05.001
81. Electronic Frontier Foundation: HTTPS Everywhere Website. https://www.eff.org/https-everywhere (2013). Accessed Apr. 2013
82. ENISA: Cloud Computing: Benefits, Risks and Recommendations for Infomarion Security. http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-risk-assessment (2009). Accessed Sept. 2012
83. Firdhous, M., Ghazali, O., Hassan, S.: A trust computing mechanism for cloud computing with multilevelthresholding. In: 6th IEEE International Conference on Industrial and Information Systems, pp. 457–461. IEEE (2011). doi:10.1109/ICIINFS.2011.6038113
84. FireEye: FireEye Advanced Threat Report—2H 2012. http://www2.fireeye.com/rs/fireye/images/fireeye-advanced-threat-report-2h2012.pdf (2013). Accessed Apr. 2013
85. Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud computing and grid computing 360-degree compared. In: Grid Computing Environments Workshop, pp. 1–10. IEEE (2008). doi:10.1109/GCE.2008.4738445
86. Garfinkel, T., Rosenblum, M.: When virtual is harder than real: security challenges in virtual machine based computing environments. In: Proceedings of the 10th Conference on Hot Topics in Operating Systems, vol. 10, pp. 20–20. USENIX Association, Berkeley, CA, USA (2005)
87. Gartner: Assessing the Security Risks of Cloud Computing. http://cloud.ctrls.in/files/assessing-the-security-risks.pdf (2008). White Paper. Accessed Sept. 2012
88. Gens, F.: IT Cloud Services User Survey, pt.2: Top Benefits & Challenges. IDC (2008)
89. Gens, F.: New IDC IT Cloud Services Survey: Top Benefits and Challenges. IDC (2009)
90. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC), STOC '09, pp. 169–178. ACM, Bethesda, MD, USA (2009). doi:10.1145/1536414.1536440
91. Geoffray, N., Thomas, G., Muller, G., Parrend, P., Frenot, S., Folliot, B.: I-JVM: a Java virtual machine for component isolation in OSGi. In: IEEE/IFIP Int. Conf. on Dependable Systems Networks (DSN), pp. 544–553. Estoril, Lisbon, Portugal (2009). doi:10.1109/DSN.2009.5270296
92. Gomathisankaran, M., Tyagi, A., Namuduri, K.: HORNS: a homomorphic encryption scheme for cloud computing using Residue number system. In: 45th Annual Conference on Information Sciences and Systems (CISS), pp. 1–5. Baltimore, MD, USA (2011). doi:10.1109/CISS.2011.5766176
93. Gong, C., Liu, J., Zhang, Q., Chen, H., Gong, Z.: The characteristics of cloud computing. In: 39th International Conference on Parallel Processing Workshop, pp. 275–279. IEEE Computer Society, Washington, DC, USA (2010). doi:10.1109/ICPPW.2010.45
94. Gonzalez, N., Miers, C., Redigolo, F., Carvalho, T., Simplicio, M., Naslund, M., Pourzandi, M.: A quantitative analysis of current security concerns and solutionsfor cloud computing. In: IEEE 3rd International Conference on Cloud Computing Technology and Science, pp. 231–238. IEEE Computer Society, Washington, DC, USA (2011).
95. Goodin, D.: Why Passwords have Never been Weaker—and Crackers have Never beenStronger. Ars Technica (2012)
96. Goodrich, R.: What Is Doxing? TechNewsDaily (2013)
97. Google: Google App Engine. https://developers.google.com/appengine/ (2013). Accessed Apr. 2013
98. Green, M.: The threat in the cloud. IEEE Secur. Priv. **11**(1), 86–89 (2013). doi:10.1109/MSP.2013.20
99. Grispos, G., Glisson, W.B., Storer, T.: Using smartphones as a proxy for forensic evidence contained incloud storage services. In: 46th Hawaii International Conference on System Sciences (HICSS), pp. 4910–4919. Maui, HI, USA (2013). doi:10.1109/HICSS.2013.592
100. Grobauer, B., Walloschek, T., Stocker, E.: Understanding cloud computing vulnerabilities. IEEE Secur. Priv. **9**(2), 50–57 (2011). doi:10.1109/MSP.2010.115
101. Grosse, E., Upadhyay, M.: Authentication at scale. IEEE Secur. Priv. **11**(1), 15–22 (2013). doi:10.1109/MSP.2012.162
102. Gruschka, N., Iacono, L.: Vulnerable cloud: SOAP message security validation revisited. In: IEEE International Conference on Web Services, pp. 625–631. IEEE Computer Society, Washington, DC, USA (2009). doi:10.1109/ICWS.2009.70
103. Gul, I., Rehman, A., Islam, M.: Cloud computing security auditing. In: The 2nd International Conference on Next Generation Information Technology, pp. 143–148. IEEE (2011)
104. Habib, S., Ries, S., Muhlhauser, M.: Towards a trust management system for cloud computing. In: IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 933–939. IEEE Computer Society, Washington, DC, USA (2011). doi:10.1109/TrustCom.2011.129
105. Hale, C.: bcrypt. http://codahale.com/how-to-safely-store-a-password/ (2010). Accessed May 2013
106. Hamada, J.: Japanese One-Click Fraud Campaign Comes to Google Play. Symantec Blog (2013)
107. Hart, J.: Remote working: managing the balancing act between network access and data security. Comput. Fraud Secur. **2009**(11), 14–17 (2009). doi:10.1016/S1361-3723(09)70141-1
108. Hayes, B.: Cloud computing. Commun. ACM **51**(7), 9–11 (2008). doi:10.1145/1364782.1364786
109. Helland, P.: Condos and clouds. Commun. ACM **56**(1), 50–59 (2013). doi:10.1145/2398356.2398374
110. Heninger, N., Durumeric, Z., Wustrow, E., Halderman, J.A.: Minding your Ps and Qs: detection of widespread weak keys in network devices. In: Proceedings of the 21st USENIX Security Symposium, pp. 205–220. USENIX, Bellevue, WA, USA (2012). doi:10.1109/ICCIAutom.2011.6183990
111. Hodges, J., Jackson, C., Barth, A.: HTTP Strict Transport Security (HSTS). RFC 6797 (Proposed Standard) (2012). https://www.ietf.org/rfc/rfc6797.txt
112. Honan, M.: How Apple and Amazon Security Flaws Led to My Epic Hacking. Wired (2012)
113. HP: HP 2012 Cyber Risk Report. http://www.hpenterprisesecurity.com/collateral/whitepaper/HP2012CyberRiskReport_0213.pdf (2013). Accessed Apr. 2013
114. HP: HP ArcSight. http://www8.hp.com/us/en/software-solutions/software.html?compURI=1340477 (2013). Accessed Apr. 2013
115. Hua, J., Sakurai, K.: Barrier: a lightweight hypervisor for protecting kernel integrityvia memory isolation. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC), pp. 1470–1477. ACM, Trento, Italy (2012). doi:10.1145/2231936.2232011
116. Hunt, T.: 5 Ways to Implement HTTPS in an Insufficient Manner (and leak sensitive data). http://www.troyhunt.com/2013/04/5-ways-to-implement-https-in.html (2013). Accessed Apr. 2013

117. Idziorek, J., Tannian, M.: Exploiting cloud utility models for profit and ruin. In: IEEE International Conference on Cloud Computing, pp. 33–40. IEEE Computer Society, Washington, DC, USA (2011)
118. Idziorek, J., Tannian, M., Jacobson, D.: Detecting fraudulent use of cloud resources. In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security, pp. 61–72. ACM, New York, NY, USA (2011). doi:10.1145/2046660.2046676
119. Infosecurity: Recycled phones retain their previous owners' data. Infosecurity Magazine (2013)
120. Intel: Intel Digital Random Number Generator (DRNG): Software Implementation Guide. http://software.intel.com/sites/default/files/m/d/4/1/d/8/441_Intel_R_DRNG_Software_Implementation_Guide_final_Aug7.pdf (2012). Accessed May 2013
121. Jackson, C.: 8 Cloud Security Concepts You Should Know. Network World (2010)
122. Jackson, C., Barth, A.: ForceHTTPS: protecting high-security web sites from network attacks. In: Proceedings of the 17th International Conference on World Wide Web (WWW), pp. 525–534. ACM, Beijing, China (2008). doi:10.1145/1367497.1367569
123. Jasti, A., Shah, P., Nagaraj, R., Pendse, R.: Security in multitenancy cloud. In: IEEE International Carnahan Conference on Security Technology, pp. 35–41. IEEE (2010). doi:10.1109/CCST.2010.5678682
124. Jenkins, Q.: Spamhaus: DDoS Update—March 2013. Spamhaus (2013)
125. Jensen, M., Gruschka, N., Herkenhöner, R.: A survey of attacks on web services. Comput. Sci. Res. Dev. **24**, 185–197 (2009). doi:10.1007/s00450-009-0092-6
126. Jensen, M., Gruschka, N., Luttenberger, N.: The impact of flooding attacks on network-based services. In: 3rd International Conference on Availability, Reliability and Security, pp. 509–513. IEEE Computer Society, Washington, DC, USA (2008)
127. Jensen, M., Meyer, C.: Expressiveness considerations of XML signatures. In: IEEE 35th Annual Computer Software and Applications Conf. Workshop, pp. 392–397. IEEE Computer Society, Washington, DC, USA (2011)
128. Jensen, M., Schäge, S., Schwenk, J.: Towards an anonymous access control and accountability scheme for cloud computing. In: IEEE 3rd International Conference on Cloud Computing, pp. 540–541. IEEE Computer Society, Washington, DC, USA (2010). doi:10.1109/CLOUD.2010.61
129. Jensen, M., Schwenk, J.: The accountability problem of flooding attacks in service-oriented architectures. In: International Conference on Availability, Reliability and Security, pp. 25–32. IEEE (2009)
130. Jensen, M., Schwenk, J., Gruschka, N., Iacono, L.: On Technical security issues in cloud computing. In: IEEE International Conference on Cloud Computing, pp. 109–116. IEEE Computer Society, Washington, DC, USA (2009). doi:10.1109/CLOUD.2009.60
131. Jin, B., Wang, Y., Liu, Z., Xue, J.: A trust model based on cloud model and Bayesian networks. Procedia Environ. Sci. **11, Part A**, 452–459 (2011). doi:10.1016/j.proenv.2011.12.072
132. Kandukuri, B., Paturi, V., Rakshit, A.: Cloud security issues. In: IEEE International Conference on Services Computing, pp. 517–520. IEEE (2009). doi:10.1109/SCC.2009.84
133. Kant, K.: Data center evolution: a tutorial on state of the art, issues, and challenges. Comput. Netw. **53**(17), 2939–2965 (2009). doi:10.1016/j.comnet.2009.10.004
134. Katsuki, T.: Crisis for Windows Sneaks onto Virtual Machines. Symantec Blog (2012)
135. Kaufman, L.: Data security in the world of cloud computing. IEEE Secur. Priv. **7**(4), 61–64 (2009)
136. Kerrigan, B., Chen, Y.: A study of entropy sources in cloud computers: random number generation on cloud hosts. In: Proceedings of the 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security (MMM-ACNS), pp. 286–298. Springer, St. Petersburg, Russia (2012). doi:10.1007/978-3-642-33704-8_24
137. Khan, K., Malluhi, Q.: Establishing trust in cloud computing. IT Prof. **12**(5), 20–27 (2010). doi:10.1109/MITP.2010.128
138. Khorshed, M.T., Ali, A.S., Wasimi, S.A.: A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing. Future Gener. Comput. Syst. **28**(6), 833–851 (2012). doi:10.1016/j.future.2012.01.006
139. King, C.I.: Intel Rdrand Instruction Revisited. http://smackerelofopinion.blogspot.co.uk/2012/10/intel-rdrand-instruction-revisited.html (2012). Accessed May 2013
140. King, S., Chen, P.: SubVirt: implementing malware with virtual machines. In: IEEE Symposium on Security and Privacy, pp. 14 pp.-327. IEEE Computer Society, Washington, DC, USA (2006). doi:10.1109/SP.2006.38
141. Kirkland, D.: Entropy (or rather the lack thereof) in OpenStack instances... and how to improve that. http://www.openstack.org/summit/san-diego-2012/openstack-summit-sessions/presentation/entropy-or-lack-thereof-in-openstack-instances (2012). Accessed May 2013
142. Kufel, L.: Security event monitoring in a distributed systems environment. IEEE Secur. Priv. **11**(1), 36–43 (2013). doi:10.1109/MSP.2012.61
143. Leder, F., Werner, T.: Know Your Enemy: Containing Conficker. http://www.honeynet.org/files/KYE-Conficker.pdf (2010). White Paper. Accessed May 2013
144. Leder, F., Werner, T.: Containing Conficker. http://net.cs.uni-bonn.de/wg/cs/applications/containing-conficker/ (2011). Accessed May 2013
145. Lee, J.H., Park, M.W., Eom, J.H., Chung, T.M.: Multi-level intrusion detection system and log management in cloud computing. In: 13th International Conference on Advanced Communication Technology, pp. 552–555. IEEE (2011)
146. Lemos, R.: Blue Security Folds Under Spammer's Wrath. SecurityFocus (2013)
147. Lenk, A., Klems, M., Nimis, J., Tai, S., Sandholm, T.: What's inside the cloud? An architectural map of the cloud landscape. In: Proceedings of the ICSE Workshop on Software Engineering Challenges of Cloud Computing, pp. 23–31. IEEE Computer Society, Washington, DC, USA (2009). doi:10.1109/CLOUD.2009.5071529
148. Leopando, J.: World Backup Day: The 3–2–1 Rule. Trend Micro TrendLabs (2013)
149. Li, F., Lai, A., Ddl, D.: Evidence of advanced persistent threat: a case study of malware for political espionage. In: 6th International Conference on Malicious and Unwanted Software (MALWARE), pp. 102–109. Fajardo, PR, USA (2011). doi:10.1109/MALWARE.2011.6112333
150. Li, H.C., Liang, P.H., Yang, J.M., Chen, S.J.: Analysis on cloud-based security vulnerability assessment. In: IEEE 7th International Conference on e-Business Engineering, pp. 490–494. IEEE (2010). doi:10.1109/ICEBE.2010.77
151. Li, Q., Clark, G.: Mobile security: a look ahead. IEEE Secur. Priv. **11**(1), 78–81 (2013). doi:10.1109/MSP.2013.15
152. Li, X., Loh, P., Tan, F.: Mechanisms of polymorphic and metamorphic viruses. In: European Intelligence and Security Informatics Conference (EISIC), pp. 149–154. Berkeley/Oakland, CA, USA (2011). doi:10.1109/EISIC.2011.77
153. Liu, F., Su, X., Liu, W., Shi, M.: The design and application of Xen-based host system firewall and its extension. In: International Conference on Electronic Computer Technology, pp. 392–395. Macau, China (2009). doi:10.1109/ICECT.2009.83
154. Liu, H.: A new form of DoS attack in a cloud and its avoidance mechanism. In: Proceedings of the ACM Workshop on Cloud

Computing Security, pp. 65–76. ACM, New York, NY, USA (2010). doi:10.1145/1866835.1866849
155. LivingSocial: LivingSocial Security Notice. https://livingsocial.com/createpassword (2013). Accessed May 2013
156. Luo, S., Lin, Z., Chen, X., Yang, Z., Chen, J.: Virtualization security for cloud computing service. In: International Conference on Cloud and Service Computing, pp. 174–179. IEEE Computer Society, Washington, DC, USA (2011)
157. Mandiant: APT1: Exposing One of China's Cyber Espionage Units. http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf (2013). Accessed Apr. 2013
158. Mansfield-Devine, S.: Danger in the clouds. Netw. Secur. **2008**(12), 9–11 (2008). doi:10.1016/S1353-4858(08)70140-5
159. Marlinspike, M.: New tricks for defeating SSL in practice. https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf (2009). Accessed Apr. 2013
160. Marlinspike, M.: sslstrip. http://www.thoughtcrime.org/software/sslstrip/ (2009). Accessed Apr. 2013
161. Martin, D.: Implementing effective controls in a mobile, agile, cloud-enabled enterprise. IEEE Secur. Priv. **11**(1), 13–14 (2013). doi:10.1109/MSP.2013.1
162. Mathisen, E.: Security challenges and solutions in cloud computing. In: Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies, pp. 208–212. IEEE (2011). doi:10.1109/DEST.2011.5936627
163. McAfee: McAfee Threats Report—Fourth Quarter 2012. http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q4-2012.pdf (2013). Accessed Apr. 2013
164. McCune, J., Li, Y., Qu, N., Zhou, Z., Datta, A., Gligor, V., Perrig, A.: TrustVisor: efficient TCB reduction and attestation. In: IEEE Symposium on Security and Privacy (SP), pp. 143–158. Oakland, CA, USA (2010). doi:10.1109/SP.2010.17
165. McGraw, G.: Software security. IEEE Secur. Priv. **2**(2), 80–83 (2004). doi:10.1109/MSECP.2004.1281254
166. McIntosh, M., Austel, P.: XML signature element wrapping attacks and countermeasures. In: Proceedings of the Workshop on Secure Web Services, pp. 20–27. ACM, New York, NY, USA (2005). doi:10.1145/1103022.1103026
167. McKendrick, J.: 7 Predictions for Cloud Computing in 2013 That Make Perfect Sense. Forbes (2012)
168. MEGA: The MEGA API. https://mega.co.nz/#developers (2013). Accessed Apr. 2013
169. Microsoft: Microsoft Hyper-V Server 2012 Website. https://www.microsoft.com/en-us/server-cloud/hyper-v-server/ (2013). Accessed Jun. 2013
170. Microsoft: Microsoft Security Intelligence Report: Volume 14. http://www.microsoft.com/security/sir/default.aspx (2013). Accessed Apr. 2013
171. Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., Rajarajan, M.: A survey of intrusion detection techniques in cloud. J. Netw. Comput. Appli. (2012). doi:10.1016/j.jnca.2012.05.003. Available online 2 June 2012
172. Mohamed, E., Abdelkader, H., El-Etriby, S.: Enhanced data security model for cloud computing. In: 8th International Conference on Informatics and Systems, pp. CC-12–CC-17. IEEE (2012)
173. Mohan, V., Hamlen, K.W.: Frankenstein: stitching malware from benign binaries. In: Proceedings of the 6th USENIX Conference on Offensive Technologies, pp. 8–8. USENIX Association, Bellevue, WA, USA (2012)
174. Monfared, A., Jaatun, M.: Monitoring intrusions and security breaches in highly distributed cloud environments. In: IEEE 3rd International Conference on Cloud Computing Technology and Science, pp. 772–777. IEEE Computer Society, Washington, DC, USA (2011). doi:10.1109/CloudCom.2011.119
175. Morsy, M.A., Grundy, J., Müller, I.: An analysis of the cloud computing security problem. In: Proceedings of Asia Pacific Software Engineering Conference Cloud Workshop, pp. 1–6. IEEE Computer Society, Washington, DC, USA (2010)
176. Moser, S.: Change I7d8c1f9b: add 'random_seed' entry to instance metadata. https://review.openstack.org/#c/14550/ (2012). Accessed May 2013
177. MPICH: MPICH Website. http://www.mpich.org/ (2013). Accessed Apr. 2013
178. Musthaler, L.: DDoS-as-a-Service? You Betcha! It's Cheap, It's Easy, and It's Available to Anyone. Security Bistro (2012)
179. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: 30th IEEE Symposium on Security and Privacy, pp. 173–187. IEEE Computer Society, Washington, DC, USA (2009). doi:10.1109/SP.2009.22
180. Nathoo, N.: Cloud Wars—The Fall of Cloud Storage. CloudTimes (2013). Accessed Apr. 2013
181. Nebula: Introducing Nebula One. https://www.nebula.com/nebula-one (2013). Accessed Apr. 2013
182. Network-Tools: Network-Tools Website. http://network-tools.com/ (2013). Accessed Apr. 2013
183. Newsome, J., Karp, B., Song, D.: Polygraph: automatically generating signatures for polymorphic worms. In: IEEE Symposium on Security and Privacy, pp. 226–241. Athens, Greece (2005). doi:10.1109/SP.2005.15
184. NIST: NIST Cloud Computing Reference Architecture. http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505 (2011). Accessed Jul. 2013
185. NIST: The NIST Definition of Cloud Computing. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (2011). Accessed Sept. 2012
186. NIST: NIST Cloud Computing Program. http://www.nist.gov/itl/cloud/ (2012). Accessed Sept. 2012
187. NIST: NIST Cloud Computing Security Reference Architecture. http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/CloudSecurity/NIST_Security_Reference_Architecture_2013.05.15_v1.0.pdf (2013). Accessed Jul. 2013
188. Oberheide, J., Cooke, E., Jahanian, F.: Empirical exploitation of live virtual machine migration. In: Proceedings of the Black Hat Convention (2008). doi:10.1109/ICCIAutom.2011.6183990
189. OCCI: OCCI Website. http://occi-wg.org/ (2013). Accessed Apr. 2013
190. Okamura, K., Oyama, Y.: Load-based covert channels between Xen virtual machines. In: Proceedings of the ACM Symposium on Applied Computing, pp. 173–180. ACM, New York, NY, USA (2010). doi:10.1145/1774088.1774125
191. O'Kane, P., Sezer, S., McLaughlin, K.: Obfuscation: the hidden malware. IEEE Secur. Priv. **9**(5), 41–47 (2011). doi:10.1109/MSP.2011.98
192. O'Neill, M.: Cloud APIs—the Next Battleground for Denial-of-Service Attacks. CSA Blog (2013)
193. Open Cloud Initiative (OCI): OCI Website. http://www.opencloudinitiative.org/ (2013). Accessed May 2013
194. OpenNebula: OpenNebula Website. http://opennebula.org/ (2013). Accessed Apr. 2013
195. OpenStack: OpenStack Website. http://www.openstack.org/ (2013). Accessed Apr. 2013
196. Oracle: Oracle Java SE Critical Patch Update Advisory—April 2013. http://www.oracle.com/technetwork/topics/security/javacpuapr2013-1928497.html (2013). Accessed Apr. 2013
197. Oracle: VirtualBox Website. https://www.virtualbox.org/ (2013). Accessed Jun. 2013
198. Ortega, A.: Your Malware Shall Not Fool Us With Those Anti Analysis Tricks. AlienVault Labs (2012)
199. OSVDB: The Open Source Vulnerability Database Website. http://www.osvdb.org/ (2013). Accessed Apr. 2013

200. OWASP: The Then Most Critical Web Application Security Risks. http://owasptop10.googlecode.com/files/OWASP (2010). Accessed Oct. 2012
201. OWASP: The Then Most Critical Web Application Security Risks. https://www.owasp.org/index.php/Top_10_2013 (2013). Accessed Apr. 2013
202. Oyama, Y., Giang, T.T.D., Chubachi, Y., Shinagawa, T., Kato, K.: Detecting malware signatures in a thin hypervisor. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC), pp. 1807–1814. ACM, Trento, Italy (2012). doi:10.1145/2231936.2232070
203. Panah, A., Panah, A., Panah, O., Fallahpour, S.: Challenges of security issues in cloud computing layers. Rep. Opin. **4**(10), 25–29 (2012)
204. Parallels: Oracle VM Server Website. http://www.oracle.com/us/technologies/virtualization/oraclevm/ (2013). Accessed Jun. 2013
205. Parallels: Parallels Website. http://www.parallels.com/eu/products/ (2013). Accessed Jun. 2013
206. Patel, A., Taghavi, M., Bakhtiyari, K., Júnior, J.C.: An intrusion detection and prevention system in cloud computing: a systematic review. J. Netw. Comput. Appli. (2012). doi:10.1016/j.jnca.2012.08.007. Available online 31 Aug. 2012
207. Patel, P.: Solution: FUTEX_WAIT hangs Java on Linux / Ubuntu in vmware or virtual box. http://www.springone2gx.com/blog/pratik_patel/2010/01/solution_futex_wait_hangs_java_on_linux_ubuntu_in_vmware_or_virtual_box(2010). Accessed May 2013
208. Patidar, S., Rane, D., Jain, P.: A survey paper on cloud computing. In: 2nd International Conference on Advanced Computing Communication Technologies, pp. 394–398. IEEE (2012). doi:10.1109/ACCT.2012.15
209. PCI Security Standards: PCI SSC Data Security Standards Overview. https://www.pcisecuritystandards.org/security_standards/index.php (2012). Accessed Oct. 2012
210. Pearce, M., Zeadally, S., Hunt, R.: Virtualization: issues, security threats, and solutions. ACM Comput. Surv. **45**(2), 1:71–1:739 (2013). doi:10.1145/2431211.2431216
211. Pearson, S.: Privacy, security and trust in cloud computing. In: Pearson, S., Yee, G. (eds.) Privacy and Security for Cloud Computing, pp. 3–42. Springer London (2013). doi:10.1007/978-1-4471-4189-1_1
212. Perez-Botero, D., Szefer, J., Lee, R.B.: Characterizing hypervisor vulnerabilities in cloud computing servers. In: Proceedings of the 2013 International Workshop on Security in Cloud Computing (SCC), pp. 3–10. ACM, New York, NY, USA (2013). doi:10.1145/2484402.2484406
213. Pfaff, B., Pettit, J., Koponen, T., Amidon, K., Casado, M., Shenker, S.: Extending networking into the virtualization layer. In: Proceedings of the 8th ACM Workshop on Hot Topics in Networks. ACM SIGCOMM (2009)
214. Prandini, M., Ramilli, M., Cerroni, W., Callegati, F.: Splitting the HTTPS stream to attack secure web connections. IEEE Secur. Priv. **8**(6), 80–84 (2010). doi:10.1109/MSP.2010.190
215. Prince, M.: The DDoS That Almost Broke the Internet. CloudFlare (2013)
216. Prince, M.: The DDoS That Knocked Spamhaus Offline (And How We Mitigated It). CloudFlare (2013)
217. Prolexic: Prolexic Quarterly Global DDoS Attack Report Q1 2013. https://www.prolexic.com/knowledge-center-ddos-attack-report-2013-q1.html (2013). Accessed Apr. 2013
218. Rahaman, M.A., Schaad, A., Rits, M.: Towards secure SOAP message exchange in a SOA. In: Proceedings of the 3rd ACM Workshop on Secure Web Services, pp. 77–84. ACM, New York, NY, USA (2006). doi:10.1145/1180367.1180382
219. Ramgovind, S., Eloff, M., Smith, E.: The management of security in cloud computing. In: Information Security for South Africa, pp. 1–7. IEEE (2010). doi:10.1109/ISSA.2010.5588290
220. Rasmusson, L., Aslam, M.: Protecting private data in the cloud. In: Proceedings of the 2nd International Conference on Cloud Computing and Services Science (CLOSER), pp. 5–12. Porto, Portugal (2012)
221. Rauti, S., Leppänen, V.: Browser extension-based man-in-the-browser attacks against Ajaxapplications with countermeasures. In: Proceedings of the 13th International Conference on Computer Systems and Technologies (CompSysTech), pp. 251–258. ACM, Ruse, Bulgaria (2012) doi:10.1145/2383276.2383314
222. RedHat: KVM Website. http://www.linux-kvm.org/ (2013). Accessed Jun. 2013
223. RepoCERT: Botnet Using Plesk Vulnerability and Takedown. Seclists Website (2013)
224. Rimal, B.P., Jukan, A., Katsaros, D., Goeleven, Y.: Architectural requirements for cloud computing systems: an enterprise cloud approach. J. Grid Comput. **9**(1), 3–26 (2011). doi:10.1007/s10723-010-9171-y
225. Ripe, NCC: Database Query. http://apps.db.ripe.net/search/query.html (2013). Accessed Apr. 2013
226. Riquet, D., Grimaud, G., Hauspie, M.: Large-scale coordinated attacks: impact on the cloud security. In: 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 558–563. IEEE (2012). doi:10.1109/IMIS.2012.76
227. Ristenpart, T., Tromer, E., Shacham, H., Savage, S.: Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 199–212. ACM, New York, NY, USA (2009)
228. Ristenpart, T., Yilek, S.: When good randomness goes bad: virtual machine reset vulnerabilities and hedging deployed cryptography. In: Proceedings of Network and Distributed Security Symposium (NDSS), pp. 1–18. The Internet Society, San Diego, CA, USA (2010)
229. Roberts II, J.C., Al-Hamdani, W.: Who can you trust in the cloud?: a review of security issues within cloud computing. In: Proceedings of the Information Security Curriculum Development Conference, pp. 15–19. ACM, New York, NY, USA (2011). doi:10.1145/2047456.2047458
230. Rocha, F., Abreu, S., Correia, M.: The final Frontier: confidentiality and privacy in the cloud. Computer **44**(9), 44–50 (2011). doi:10.1109/MC.2011.223
231. Rocha, F., Correia, M.: Lucy in the sky without diamonds: stealing confidential data in the cloud. In: IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops, pp. 129–134. IEEE (2011). doi:10.1109/DSNW.2011.5958798
232. Rodero-Merino, L., Vaquero, L.M., Caron, E., Desprez, F., Muresan, A.: Building safe PaaS clouds: a survey on security in multitenant software platforms. Comput. Secur. **31**(1), 96–108 (2012). doi:10.1016/j.cose.2011.10.006
233. Rong, C., Nguyen, S.T., Jaatun, M.G.: Beyond lightning: a survey on security challenges in cloud computing. Comput. Electr. Eng. (2012). doi:10.1016/j.compeleceng.2012.04.015 Available online 19 May 2012
234. Roy, I., Setty, S.T.V., Kilzer, A., Shmatikov, V., Witchel, E.: Airavat: security and privacy for MapReduce. In: Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation, pp. 20–20. USENIX Association, Berkeley, CA, USA (2010)
235. RSA: RSA SecurID Website. http://sweden.emc.com/security/rsa-securid.htm (2013). Accessed Jun. 2013

236. RSA FirstWatch: Tales from the Darkside: Another Mule Recruitment Site. RSA Blog (2013)
237. Rutkowska, J.: Subverting VistaTM Kernel for fun and profit. Black Hat Conv. (2008)
238. Sabahi, F.: Cloud computing security threats and responses. In: IEEE 3rd International Conference on Communication Software and Networks, pp. 245–249. IEEE (2011). doi:10.1109/ICCSN.2011.6014715
239. Sadashiv, N., Kumar, S.: Cluster, grid and cloud computing: a detailed comparison. In: 6th International Conference on Computer Science Education, pp. 477–482. IEEE (2011). doi:10.1109/ICCSE.2011.6028683
240. Salah, K., Alcaraz, Calero J.: Using cloud computing to implement a security overlay network. IEEE Secur. Priv. **11**(1), 44–53 (2013). doi:10.1109/MSP.2012.88
241. SAML v2.0: OASIS Website. https://www.oasis-open.org/standards#samlv2.0 (2005). Accessed Apr. 2013
242. Santos, N., Gummadi, K.P., Rodrigues, R.: Towards trusted cloud computing. In: Proceedings of the Conference on Hot Topics in Cloud Computing. USENIX Association, Berkeley, CA, USA (2009)
243. Schloesser, M., Guarnieri, C.: Vaccinating Systems Against VM-aware Malware. Rapid7 Labs (2013)
244. Schloesser, M., Guarnieri, C.: Vaccinating Systems Against VM-aware Malware. https://github.com/rapid7/vaccination (2013). Accessed May 2013
245. Schneier, B.: Homomorphic Encryption Breakthrough. https://www.schneier.com/blog/archives/2009/07/homomorphic_enc.html (2009). Accessed May 2013
246. SecurityFocus: Xen CVE-2013-1920 Local Memory Corruption Vulnerability. SecurityFocus (2013)
247. Sekar, V., Maniatis, P.: Verifiable resource accounting for cloud computing services. In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security, pp. 21–26. ACM, New York, NY, USA (2011). doi:10.1145/2046660.2046666
248. Sengupta, S., Kaulgud, V., Sharma, V.: Cloud computing security—trends and research directions. In: IEEE World Congress on Services, pp. 524–531. IEEE Computer Society, Washington, DC, USA (2011). doi:10.1109/SERVICES.2011.20
249. Shin, S., Gu, G.: CloudWatcher: network security monitoring using OpenFlow in dynamic cloud networks (or: how to provide security monitoring as a service in clouds?). In: 20th IEEE International Conference on Network Protocols (ICNP), pp. 1–6. Austin, TX, USA (2012).doi:10.1109/ICNP.2012.6459946
250. Shinotsuka, H.: Malware Authors Using New Techniques to Evade Automated Threat Analysis Systems. Symantec Blog (2012)
251. Singh, A.: Don't Click the Left Mouse Button: Introducing Trojan UpClicker. FireEye Blog (2012)
252. Sloan, K.: Security in a virtualised world. Netw. Secur. **2009**(8), 15–18 (2009). doi:10.1016/S1353-4858(09)70077-7
253. SNIA: Cloud Data Management Interface (CDMI). http://www.snia.org/cdmi (2013). Accessed Apr. 2013
254. Somorovsky, J., Mayer, A., Schwenk, J., Kampmann, M., Jensen, M.: On breaking SAML: be whoever you want to be. In: Proceedings of the 21st USENIX Security Symposium, pp. 21–21. USENIX Association, Bellevue, WA, USA (2012)
255. Songjie, Yao, J., Wu, C.: Cloud computing and its key techniques. In: International Conference on Electronic and Mechanical Engineering and Information Technology, vol. 1, pp. 320–324. IEEE (2011). doi:10.1109/EMEIT.2011.6022935
256. Sood, A., Enbody, R.: Targeted cyberattacks: a superset of advanced persistent threats. IEEE Secur. Priv. **11**(1), 54–61 (2013). doi:10.1109/MSP.2012.90
257. Sood, S.K.: A combined approach to ensure data security in cloud computing. J. Netw. Comput. Appli. **35**(6), 1831–1838 (2012). doi:10.1016/j.jnca.2012.07.007
258. Spoon Website: Browser Sandbox. http://spoon.net/browsers (2013). Accessed Apr. 2013
259. Stamos, A., Becherer, A., Wilcox, N.: Cloud Computing Security: Raining on the Trendy New Parade. https://www.blackhat.com/html/bh-usa-09/bh-usa-09-archives.html (2009)
260. Staten, J.: 2013 Cloud Predictions: We'll Finally Get Real About Cloud. Forrester Blog (2012)
261. Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. J. Netw. Comput. Appli. **34**(1), 1–11 (2011). doi:10.1016/j.jnca.2010.07.006
262. Sun, D., Chang, G., Sun, L., Wang, X.: Surveying and analyzing security, privacy and trust issues in cloud computing environments. Procedia Eng. **15**, 2852–2856 (2011). doi:10.1016/j.proeng.2011.08.537
263. Sun, K., Li, Y., Hogstrom, M., Chen, Y.: Sizing multi-space in heap for application isolation. In: Companion to the 21st ACM SIGPLAN Symposium on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), pp. 647–648. ACM, Portland, OR, USA (2006). doi:10.1145/1176617.1176654
264. Sun, M.K., Lin, M.J., Chang, M., Laih, C.S., Lin, H.T.: Malware virtualization-resistant behavior detection. In: IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS), pp. 912–917. Tainan, Taiwan (2011). doi:10.1109/ICPADS.2011.78
265. Suzaki, K., Iijima, K., Yagi, T., Artho, C.: Memory deduplication as a threat to the guest OS. In: Proceedings of the 4th European Workshop on System Security, pp. 1:1–1:6. ACM, Salzburg, Austria (2011). doi:10.1145/1972551.1972552
266. Suzaki, K., Iijima, K., Yagi, T., Artho, C.: Software side channel attack on memory deduplication. In: 23rd ACM Symposium on Operating Systems Principles. ACM, Cascais, Portugal (2011). Poster
267. Symantec: Internet Security Threat Report 2013. https://www.symantec.com/security_response/publications/threatreport.jsp (2013). Accessed Apr. 2013
268. Symantec Security Response: Internet Explorer Zero-Day Used in Watering Hole Attack: Q&A. Symantec Blog (2012)
269. Szefer, J., Keller, E., Lee, R.B., Rexford, J.: Eliminating the hypervisor attack surface for a more secure cloud. In: Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS), pp. 401–412. ACM, Chicago, IL, USA (2011). doi:10.1145/2046707.2046754
270. Takabi, H., Joshi, J., Ahn, G.: Security and privacy challenges in cloud computing environments. IEEE Secur. Priv. **8**(6), 24–31 (2010)
271. Tang, M., Lv, Q., Lu, Z., Zhao, Q., Song, Y.: Dynamic virtual switch protocol using Openflow. In: 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel Distributed Computing (SNPD), pp. 603–608. Kyoto, Japan (2012). doi:10.1109/SNPD.2012.129
272. Tanvi: Mixed Content Blocking Enabled in Firefox 23! Firefox Blog (2013)
273. Taylor, G., Cox, G.: Digital randomness. IEEE Spectr. **48**(9), 32–58 (2011). doi:10.1109/MSPEC.2011.5995897
274. Taylor, M., Haggerty, J., Gresty, D., Lamb, D.: Forensic investigation of cloud computing systems. Netw. Secur. **2011**(3), 4–10 (2011). doi:10.1016/S1353-4858(11)70024-1
275. The Linux Foundation: Xen Website. http://http://www.xenproject.org/ (2013). Accessed Jun. 2013
276. Thompson, H.: The human element of information security. IEEE Secur. Priv. **11**(1), 32–35 (2013). doi:10.1109/MSP.2012.161

277. Thorsheim, P.: The Final Word on the LinkedIn Leak. http://securitynirvana.blogspot.pt/2012/06/final-word-on-linkedin-leak.html (2012). Accessed May 2013
278. Toubiana, V., Nissenbaum, H.: Analysis of Google logs retention policies. J. Priv. Confid. **3**(1), 3–26 (2011)
279. Townsend, M.: Managing a security program in a cloud computing environment. In: Information Security Curriculum Development Conference, pp. 128–133. ACM, New York, NY, USA (2009). doi:10.1145/1940976.1941001
280. Trader, T.: GPU Monster Shreds Password Hashes. HPCwire (2012)
281. Tripathi, A., Mishra, A.: Cloud computing security considerations. In: IEEE International Conference on Signal Processing, Communications and Computing, pp. 1–5. IEEE (2011). doi:10.1109/ICSPCC.2011.6061557
282. Tsai, H.Y., Siebenhaar, M., Miede, A., Huang, Y., Steinmetz, R.: Threat as a service?: virtualization's impact on cloud security. IT Prof. **14**(1), 32–37 (2012). doi:10.1109/MITP.2011.117
283. Tseng, H.M., Lee, H.L., Hu, J.W., Liu, T.L., Chang, J.G., Huang, W.C.: Network virtualization with cloud virtual switch. In: IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS), pp. 998–1003. Tainan, Taiwan (2011). doi:10.1109/ICPADS.2011.159
284. Vaquero, L.M., Rodero-Merino, L., Morán, D.: Locking the sky: a survey on IaaS cloud security. Computing **91**(1), 93–118 (2011). doi:10.1007/s00607-010-0140-x
285. Viega, J.: Cloud computing and the common man. Computer **42**(8), 106–108 (2009). doi:10.1109/MC.2009.252
286. VMware: VMware vSphere. https://www.vmware.com/support/product-support/vsphere/ (2013). Accessed Apr. 2013
287. VMware: VMware Website. https://www.vmware.com/products/ (2013). Accessed Jun. 2013
288. VMware: What is OVF? https://www.vmware.com/technical-resources/virtualization-topics/virtual-appliances/ovf.html (2013). Accessed Apr. 2013
289. VMware Community Forums: Low/proc/sys/kernel/random/entropy_avail causes exim to stop sending mail. http://communities.vmware.com/message/530909 (2006). Accessed May 2013
290. Vu, Q.H., Pham, T.V., Truong, H.L., Dustdar, S., Asal, R.: DEMODS: a description model for data-as-a-service. In: IEEE 26th International Conference on Advanced Information Networking and Applications (AINA), pp. 605–612. Fukuoka, Japan (2012). doi:10.1109/AINA.2012.91
291. Wang, C., Ren, K., Lou, W., Li, J.: Toward publicly auditable secure cloud data storage services. IEEE Netw. **24**(4), 19–24 (2010). doi:10.1109/MNET.2010.5510914
292. Wang, C., Wang, Q., Ren, K., Lou, W.: Ensuring data storage security in cloud computing. In: 17th International Workshop on Quality of Service, pp. 1–9. IEEE (2009). doi:10.1109/IWQoS.2009.5201385
293. Wang, G., Ng, T.: The impact of virtualization on network performance of Amazon EC2 data center. In: Proceedings of the IEEE INFOCOM, pp. 1–9. Sand Diego, CA, USA (2010). doi:10.1109/INFCOM.2010.5461931
294. Ward, M.: Facebook Users Suffer Viral Surge. BBC News (2009)
295. Websense: 2013 Threat Report. https://www.websense.com/content/websense-2013-threat-report.aspx (2013). Accessed Apr. 2013
296. Wei, J., Zhang, X., Ammons, G., Bala, V., Ning, P.: Managing security of virtual machine images in a cloud environment. In: Proceedings of the ACM Workshop on Cloud Computing Security, pp. 91–96. ACM, New York, NY, USA (2009). doi:10.1145/1655008.1655021
297. Wu, H., Ding, Y., Winer, C., Yao, L.: Network security for virtual machine in cloud computing. In: 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT), pp. 18–21. Seoul, South Korea (2010). doi:10.1109/ICCIT.2010.5711022
298. Wu, H., Ding, Y., Winer, C., Yao, L.: Network security for virtual machine in cloud computing. In: 5th International Conference on Computer Sciences and Convergence Information Technology, pp. 18–21. IEEE (2010). doi:10.1109/ICCIT.2010.5711022
299. Wueest, C.: Mobile Scam: Winning Without Playing. Symantec Blog (2013)
300. Xiao, Z., Xiao, Y.: Security and privacy in cloud computing. IEEE Commun. Surv. Tuts. **15**(2), 843–859 (2013). doi:10.1109/SURV.2012.060912.00182
301. Xu, Y., Bailey, M., Jahanian, F., Joshi, K., Hiltunen, M., Schlichting, R.: An exploration of L2 cache covert channels in virtualized environments. In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security, pp. 29–40. ACM, New York, NY, USA (2011). doi:10.1145/2046660.2046670
302. Yang, J., Chen, Z.: Cloud computing research and security issues. In: International Conference on Computational Intelligence and Software Engineering, pp. 1–3. IEEE (2010). doi:10.1109/CISE.2010.5677076
303. Yasinsac, A., Irvine, C.: Help! Is There a Trustworthy-Systems Doctor in the House? IEEE Secur. Priv. **11**(1), 73–77 (2013). doi:10.1109/MSP.2013.10
304. Yilek, S.: Resettable public-key encryption: how to encrypt on a virtual machine. In: Proceedings of the International Conference on Topics in Cryptology, CT-RSA'10, pp. 41–56. Springer-Verlag, San Francisco, CA, USA (2010). doi:10.1007/978-3-642-11925-5_4
305. Yu, A., Sathanur, A., Jandhyala, V.: A partial homomorphic encryption scheme for secure design automation on public clouds. In: IEEE 21st Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), pp. 177–180. Tempe, AZ, USA (2012). doi:10.1109/EPEPS.2012.6457871
306. Yu, H., Powell, N., Stembridge, D., Yuan, X.: Cloud computing and security challenges. In: Proceedings of the 50th Annual Southeast Regional Conference, pp. 298–302. ACM, New York, NY, USA (2012). doi:10.1145/2184512.2184581
307. Zabidi, M., Maarof, M., Zainal, A.: Malware analysis with multiple features. In: UKSim 14th International Conference on Computer Modelling and Simulation, pp. 231–235. Cambridge, London (2012). doi:10.1109/UKSim.2012.40
308. Zhang, F., Huang, Y., Wang, H., Chen, H., Zang, B.: PALM: security preserving VM live migration for systems with VMM-enforced protection. In: 3rd Asia-Pacific Trusted Infrastructure Technologies Conference, pp. 9–18. IEEE Computer Society, Washington, DC, USA (2008). doi:10.1109/APTC.2008.15
309. Zhang, Y., Juels, A., Reiter, M.K., Ristenpart, T.: Cross-VM side channels and their use to extract private keys. In: Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS), pp. 305–316. ACM, Raleigh, NC, USA (2012). doi:10.1145/2382196.2382230
310. Zhou, M., Zhang, R., Xie, W., Qian, W., Zhou, A.: Security and privacy in cloud computing: a survey. In: 6th International Conference on Semantics Knowledge and Grid, pp. 105–112. IEEE Computer Society, Washington, DC, USA (2010)
311. Zieg, M.: Separating fact from fiction in cloud computing. Data Center J. (2012)
312. Zissis, D., Lekkas, D.: Addressing cloud computing security issues. Future Gener. Comput. Syst. **28**(3), 583–592 (2010). doi:10.1016/j.future.2010.12.006
313. Zou, B., Zhang, H.: Toward enhancing trust in cloud computing environment. In: 2nd International Conference on Control, Instrumentation and Automation, pp. 364–366 (2011). doi:10.1109/ICCIAutom.2011.6183990