Journal of Network and Computer Applications 71 (2016) 11–29

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

A survey of security issues for cloud computing

Minhaj Ahmad Khan

Bahauddin Zakariya University Multan, Pakistan

ARTICLE INFO

Article history:
Received 6 July 2015
Received in revised form
12 February 2016
Accepted 14 May 2016
Available online 21 May 2016

Keywords:
Cloud security
Cloud computing
Denial-of-service
Security threats
Intrusion detection systems

ABSTRACT

High quality computing services with reduced cost and improved performance have made cloud computing a popular paradigm. Due to its flexible infrastructure, net centric approach and ease of access, the cloud computing has become prevalent. Its widespread usage is however being diminished by the fact that the cloud computing paradigm is yet unable to address security issues which may in turn aggravate the quality of service as well as the privacy of customers' data.

In this paper, we present a survey of security issues in terms of security threats and their remediations. The contribution aims at the analysis and categorization of working mechanisms of the main security issues and the possible solutions that exist in the literature. We perform a parametric comparison of the threats being faced by cloud platforms. Moreover, we compare various intrusion detection and prevention frameworks being used to address security issues. The trusted cloud computing and mechanisms for regulating security compliance among cloud service providers are also analyzed. Since the security mechanisms continue to evolve, we also present the future orientation of cloud security issues and their possible countermeasures.

Contents

E-mail address: mik@bzu.edu.pk

http://dx.doi.org/10.1016/j.jnca.2016.05.010
1084-8045/© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Cloud computing has gained wide acceptance for organizations as well as individuals by introducing computation, storage and software based services. It is used to address the resource scarcity issues of its clients by providing them with on-demand pay-per-use services (Buyya et al., 2011). It incorporates a centralized collection of resources called a *cloud* connected through a high speed network. The global availability of high performance resources, support of a large number of services, and ability to store large amount of data have made it ubiquitous. Even with the modern smartphones, the cloud computing is able to serve multiple purposes ranging from backup of contacts to the execution of complex applications through computation offloading (Sanaei et al., 2014; Kumar et al., 2013). Moreover, the reduced cost of services and an assurance regarding quality make it an attractive solution for mitigating the issue of constrained resources. Since a cloud computing platform provides services by sharing valuable resources, an adequate usage of these resources may be achieved by ensuring that the platform is able to counter security threats which may otherwise deteriorate its performance and reliability.

An overview of a public cloud computing platform is shown in Fig. 1. The cloud platform is usually equipped with high performance server machines, high speed storage devices and an

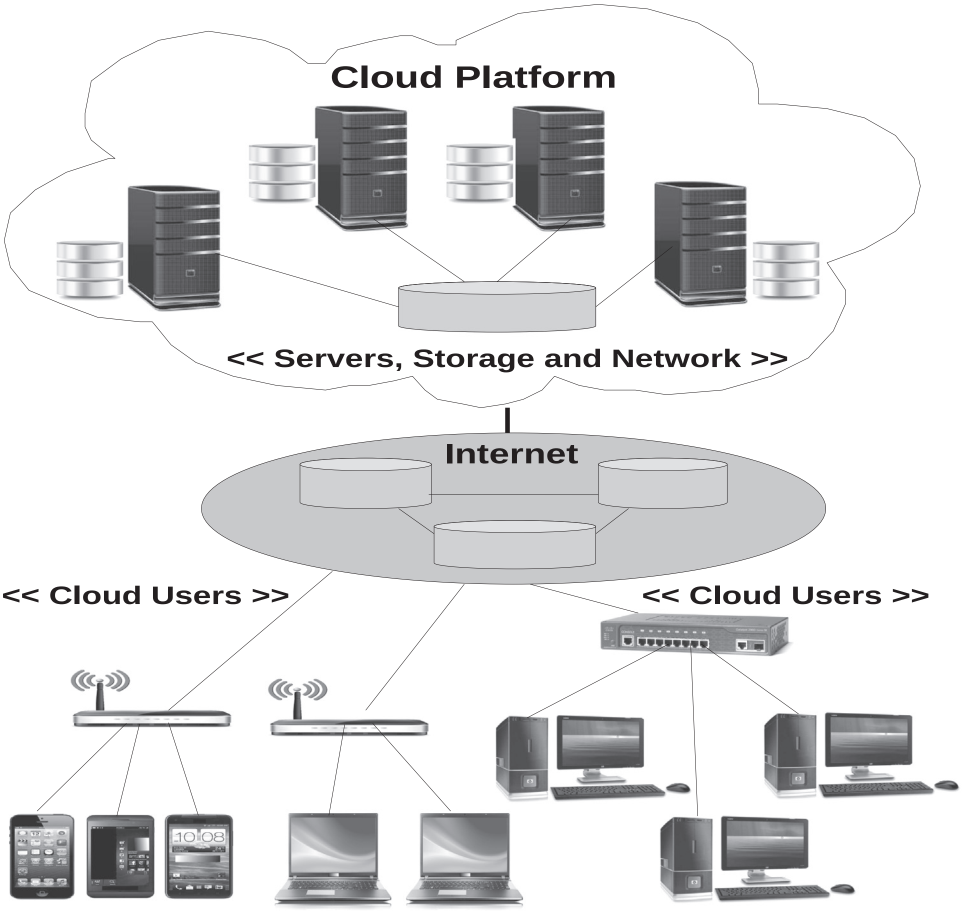

Fig. 1. Cloud computing architecture with cloud users connecting to a public cloud platform through internet.

efficient network. The cloud users having mobile phones, laptops or modern desktops connect to the cloud platform through internet. Since the server machines are connected using an internal network, an attack on the network may produce a detrimental impact in the form of communication delays or even the network being inaccessible (McMillan, 2009; InfoSecurity, 2009). Likewise, the attacks on virtual machines and hypervisors being used to run virtual machines have shown to severely breach the security for malicious purposes (Grobauer et al., 2011; Liu et al., 2014; Zhang et al., 2012; Zhou et al., 2011). Similarly, the cloud services are also prone to security threats as this layer contains software which has always been vulnerable to hacks and security attacks (Gruschka and Iacono, 2009). These attacks may cause violation of data protection or even unavailability of services for all the clients.

As the cloud computing hinges on the traditional architecture, it is becoming more vulnerable to security breaches. The Cloud Security Alliance report (Ko et al., 2013) reveals a manifold increase in the frequency of cloud outage taking place within recent years. A large number of vulnerability incidents occurred with threats already known to exist. Similarly, as per threat report published by Symantec (2015), there has been a 91% increase in targeted attacks with 1 out of 3 *Symantec.cloud* customers being targeted by spear-phishing attacks in year 2013. To cope with the increasing number of security threats, there has been a parallel advancement of countermeasures. For attacks emerging due to network such as botnet and stepping-stone attacks (McMillan, 2009; InfoSecurity, 2009), various countermeasures (Lin and Lee, 2012; Kourai et al., 2012; Wu et al., 2010) are able to detect and fix them. The risk of violation of data protection attacks (Ashford, 2015; Ristenpart et al., 2009) has been mitigated using cryptographic techniques (Wang et al., 2009; Wylie et al., 2001). The VM and hypervisor based vulnerabilities (Aviram et al., 2010; Hlavacs et al., 2011) are tackled using authentication mechanisms (Godfrey and Zulkernine, 2013; Osvik et al., 2006). Similarly, the attacks related to denial or theft-of-service (Riquet et al., 2012; Zhou et al., 2011) may be coped with the intrusion detection systems (Scarfone and Mell, 2007; Roesch, 2014; Hay et al., 2008). For the threats resulting in disclosure of information to third party, the privacy Acts such as ECPA and HIPAA (DHS, 2013; U.D. of Health & Human Services, 2007) have been deployed.

Various contributions presenting a survey of cloud security issues and challenges have been made recently. A classification of security issues while mapping threats to countermeasures is described in Khalil et al. (2014), Hashizume et al. (2013), Ashktorab and Taghizadeh (2012), and Shankarwar and Pawar (2015). Similarly, corresponding to the infrastructure, platform and software layers of the cloud computing model, a layer-wise categorization of security threats is presented in Subashini and Kavitha (2011). Another classification of attacks on clouds using attack surfaces comprising the users, services and clouds with six possible interfaces is described in Gruschka and Jensen (2010). Bisong and Rahman (2011), Khorshed et al. (2012), and Srinivasamurthy et al. (2013) discuss major cloud security threats as identified by Cloud Security Alliance (2010), whereas the confidentiality, integrity, availability, audit and control are described as major cloud security issues together with data privacy Acts in Zhou et al. (2010). Shahzad (2014) discusses malicious insider based denial-of-service attacks together with the attacks on data while using Amazon Web Services (AWS) as a case study. A brief survey given in Curran and Carlin (2011) describes the scalability of cloud and compliance regulations as major cloud security issues. The authors urge the need to build confidence of cloud customers by revealing implementation details and ensuring security compliance. The survey contribution (Modi et al., 2013) discusses various intrusion detection techniques, whereas a security analysis of open source cloud software platforms is provided in Popovic et al. (2011). In contrast to these contributions, we present a comprehensive survey with a parametric analysis of cloud security issues, countermeasures, and security frameworks (intrusion detection and prevention systems) in this paper. We employ a component based categorization in terms of the network, VM, storage and applications executing on a cloud platform. Various solutions to trusted cloud computing are also analyzed together with compliance issues in terms of prevailing standards, Acts and regulations. We also present succinctly the future orientation of security challenges and their possible solutions.

The remaining part of this paper is organized as follows. Section 2 describes a categorization of security threats and their implications. A comparative analysis of security threats and their countermeasures is given in Section 3. Section 4 presents a comparison of the intrusion detection and prevention systems aimed at providing security for the clouds. An analysis of various approaches used for trusted cloud computing is given in Section 5. Various Acts and official rules used for regulating cloud computing security issues are discussed in Section 6. The future challenges and issues together with the suggestions to cope with these issues are discussed in Section 7 before concluding the paper in Section 8.

2. Cloud computing security: taxonomy and categorization

Cloud computing offers services using the Infrastructure as a Service (*IaaS*), Platform as a Service (*PaaS*) and Software as a Service (*SaaS*) models (Buyya et al., 2011) as shown in Fig. 2. Cloud users have access to servers and virtual machines through the *IaaS* service model. The hypervisors execute on the servers to provide virtualization of physical resources. Similarly, using the *PaaS* service model, the cloud platform provides support of operating systems, runtime systems, databases or web servers. The *SaaS* service model provides support of pay-per-use software. The diversity of these service delivery models makes the cloud computing platforms more vulnerable to attacks than any other computing platform. Its vulnerability may be exposed through any of its core components: network, virtual machines, storage and applications, which are used as a basis for categorization of attacks and their implications.

2.1. Categorization of attacks based on cloud components

For performing comparative analysis, we categorize attacks on

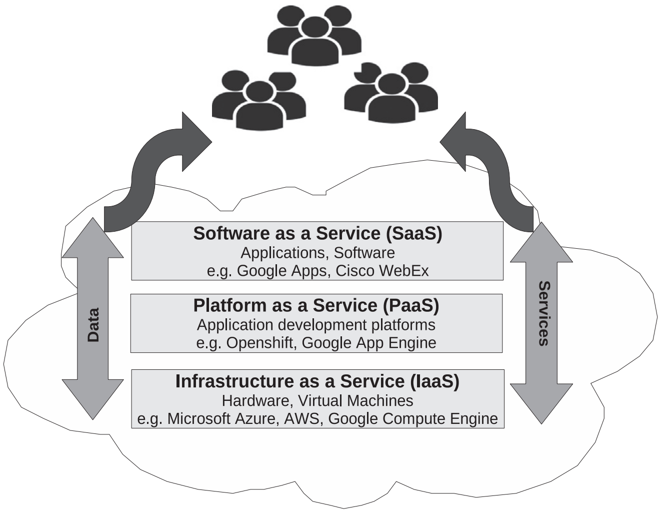
Software as a Service (SaaS)
Applications, Software
e.g. Google Apps, Cisco WebEx
Platform as a Service (PaaS)
Application development platforms
e.g. Openshift, Google App Engine
Infrastructure as a Service (IaaS)
Hardware, Virtual Machines
e.g. Microsoft Azure, AWS, Google Compute Engine
Data
Services

Fig. 2. Cloud service delivery architecture.

a cloud platform based on its components: network (A_1), virtual machines (A_2), storage (A_3) and applications (A_4) as elaborated below.

2.1.1. Network based attacks (A_1)

The cloud machines existing within a cloud platform are connected through a network which also provides connections with the machines outside the cloud platform. An intruder may attack a cloud system through its network which may in turn deteriorate the quality of cloud services and may even put data privacy/confidentiality at risk. For our analysis, we consider three types of network based attacks as elaborated below.

Port scanning (A_{1a}): A port on a server may be probed to check the status of a service executing on the target machine. The port scanning requires access to the network hosting the target machine. It is used to expose vulnerabilities of the target machine resulting in the denial-of-service (Riquet et al., 2012). The intrusion detection systems or firewalls (Scarfone and Mell, 2007; Roesch, 2014; OWASP, 2015a; Debar et al., 2007) may be used to detect and hinder such attacks.

Botnets (A_{1b}): A botnet may be used to steal data from a host machine and communicate it to a bot-master. A command and control system is established with a bot-master and several machines can act as stepping-stone to steal private information. Several incidents have shown to incorporate clouds as command and control servers (McMillan, 2009; InfoSecurity, 2009). The techniques to counter botnet attacks mainly attempt to track the botmaster by interpreting the communication or filtering the packets (Lin and Lee, 2012; Kourai et al., 2012).

Spoofing attacks (A_{1c}): The spoofing attacks in a network impersonate entities for malicious purposes. An IP spoofing attack may replace the IP address in a network packet with a forged source IP address. Similarly, a DNS spoofing attack may cause a DNS server to return an incorrect IP address thereby redirecting network traffic to an attacker's system. A virtual network may be a victim of ARP spoofing (Wu et al., 2010) thereby causing an attacker VM to access packets of other VMs. The cloud based antivirus (Oberheide et al., 2008a) or intrusion detection systems (Scarfone and Mell, 2007; Roesch, 2014; OISF, 2015) may be used to cope with these attacks.

2.1.2. VM based attacks (A_2)

On a cloud system, the VM based attacks exploit vulnerabilities in the virtual machines to violate data protection and affect the cloud services. Multiple virtual machines being hosted on a system cause several security risks. Moreover, various stages of VM management may be used to launch a large number of cloud attacks. For our analysis, we consider four types of VM based attacks as described below:

Cross VM side channel attacks (A_{2a}): The VM based side channel attacks are able to extract information regarding resource usage, cryptographic keys and other information from a target VM which is residing on the same physical machine as that of the attacker VM (Tandon et al., 2014; Zhang et al., 2012; Ristenpart et al., 2009; Hlavacs et al., 2011). These attacks may exploit timing information from resources such as cache and shared memory. The countermeasures for side channel attacks use authentication mechanisms, cryptographic algorithms or deterministic execution to mitigate the risk of side channels (Godfrey and Zulkernine, 2013; Osvik et al., 2006; Zhang et al., 2012; Aviram et al., 2010; Wang and Lee, 2007).

VM creation attacks (A_{2b}): A malicious code can be placed inside a VM image which is then replicated during creation of virtual machines (Grobauer et al., 2011; Morsy et al., 2010; Garfinkel and Rosenblum, 2005). In this regard, virtual image management system providing filters and scanners for detecting and recovering from security violations may be used (Wei et al., 2009; Reimer et al., 2008; Fernandez et al., 2013).

VM migration and rollback attacks (A_{2c}): When an active VM is being migrated from the host physical machine to another physical machine, the contents of the VM files become vulnerable to various attacks (Oberheide et al., 2008b; Garfinkel and Rosenblum, 2005). For example, the log of execution state being maintained for implementing a rollback may become accessible during migration. An effective configuration of security policies or proper suspend/resume activities may render the VM migration to be more secure (Xiaopeng et al., 2010; Santos et al., 2009; Zhang et al., 2008; Szefer and Lee, 2012).

VM scheduler based attacks (A_{2d}): A few vulnerabilities of scheduler may result in resource stealing or theft-of-service (Zhou et al., 2011; Rong et al., 2013). For example, a VM may be scheduled to run after a specific time while retaining the credit balance of the VM execution time slice. Modified versions of scheduler (Zhou et al., 2011; Gruschka and Jensen, 2010) may improve security of hypervisors while maintaining fairness and efficiency.

2.1.3. Storage based attacks (A_3)

An attacker from outside or even a malicious insider may steal private data stored on some storage device (Stefanov and Shi, 2013; Li et al., 2013; Jung et al., 2013; Cloud Security Alliance, 2012). With access to sensitive information, a large number of vulnerabilities may be exploited by manipulating data if a strict monitoring mechanism is not implemented. For analysis, we consider two types of storage based attacks on clouds as elaborated below:

Data scavenging (A_{3a}): While erasing data from a storage device, the file systems do not remove data completely. Consequently, the removed data may be recovered by attackers which is referred to as data scavenging (Hashizume et al., 2013; Jansen, 2011; Ertaul et al., 2010; Grobauer et al., 2011; Sen, 2013; Townsend, 2009). Various techniques to counter data scavenging are reported in AWS (2014).

Data deduplication (A_{3b}): With data deduplication being performed for minimizing storage and bandwidth requirements, it becomes possible to identify the files and their contents (Harnik et al., 2010). It may even make it possible to create communication channel for access to a malicious software. The deduplication exploitation risk may be mitigated by ensuring the deduplication to occur only if there is a specific number of copies of the file (Kaaniche and Laurent, 2014).

2.1.4. Application based attacks (A_4)

The applications running on a cloud may be exposed to various attacks by injecting code which may trace execution paths and exploit this information for malicious purposes. Similarly, the protocols implemented to provide services on a cloud system are vulnerable to attacks and any running applications may use them as a source of intrusion. Moreover, on a cloud system, the architectural components being shared may be exploited by an application as a source for performing malicious activities. Overall, we consider three types of application based attacks as described below:

Malware injection and steganography attacks (A_{4a}): A malicious code may be inserted in an application if a cloud platform allows for an insecure interface for application development (Ko et al., 2013; Owens, 2010). With a steganography attack, the attackers embed malicious code within files being transmitted over network (Mazurczyk and Szczypiorski, 2011). The transmission of malicious code may then be ignored by security systems for which it seems as if a normal file is being sent. The schemes like StegAD (Liu et al., 2011) may be used to identify the files and possible locations of injected code in steganographied files.

Shared architectures (A_{4b}): On a shared architecture, the execution path of a victim's application can be traced. It can be further used to detect victim's activities and hijack his account (Zhang et al., 2014). To detect the possibility of being exploited by shared architectures, the application binary code may be analyzed (Doychev et al., 2013).

Web services & protocol based attacks (A_{4c}): The web services use various protocols such as SOAP whose message header can be manipulated to contain invalid requests (Gruschka and Iacono, 2009). The security policies and validation mechanisms may be implemented to ensure valid requests for smooth running of services. To cope with possible website based threats, application firewalls may be used to identify and block the attacks (OWASP, 2015a).

2.2. Implications of attacks

An attack on a cloud may have one or more implications which may deteriorate the provision of data and services on a cloud platform. These implications are categorized as follows:

Violation of data protection: Data protection is violated when data becomes accessible to users other than owners of data. A large number of threats may violate data protection through different techniques such as data deduplication or third-party clouds (Somani et al., 2010a; Tebaa et al., 2012; Xiao and Gong, 2010; Townsend, 2009; Winkler, 2011).

Malicious manipulation of data: On cloud computing platforms, the communication between the user and the cloud services interface involves protocols such as HTTP & SOAP together with scripting languages which are vulnerable to a large number of threats (OWASP, 2015b; Fong and Okun, 2007; Karnwal et al., 2012; Gruschka and Iacono, 2009; Zhang et al., 2014). Consequently, an attacker may exploit loopholes in these mechanisms which may result in malicious manipulation of website data.

Denial-of-service: An attacker may target the cloud platform to hinder the services being provided to customers (Riquet et al., 2012; Karnwal et al., 2012). For instance, it is possible for a malicious insider to occupy the resources so that the requests by other users are responded with unavailability of resources.

Theft-of-service: A few vulnerabilities in scheduler may result in theft-of-service attacks. For example, an attacker may target the scheduling policy to be able to steal resources or obtain cloud services without proper billing (Zhou et al., 2011).

3. Comparative analysis of attacks and countermeasures

A cloud computing platform provides services using its service delivery model. The attacks on a cloud platform may exploit various components at every layer of its service model in order to violate data protection and deteriorate the quality of service for malicious purposes. This section discusses and analyzes research work contributing towards revealing attacks on clouds and their countermeasures. For a parametric evaluation, Tables 1, 2 , 3 and 4 present respectively a comparative analysis of attacks based on the network, VM, storage and application categories. The comparison is performed in terms of the attack categories, mechanisms, implications, vulnerable components and contributions for relevant countermeasures.

3.1. Network based attacks and countermeasures

Through network based attacks, the botnets have been successful in exploiting the cloud infrastructure for malicious purposes. For instance, the well known Zeus botnet was revealed to be using Amazon's Elastic Computing Cloud (EC2) as command and control server (McMillan, 2009). The Zeus botnet is a malware which is able to steal passwords. Its variants have been reported to be used for illegal bank transactions. Similarly, the *Google* cloud platform *AppEngine* has been used as a botnet to communicate with the infected computers (InfoSecurity, 2009).

A distributed denial-of-service attack (Riquet et al., 2012) performs a distributed portscan which is very difficult to be detected by an Intrusion Detection System (IDS) (Scarfone and Mell, 2007). For simulation of the attack, two configurations of a cloud platform secured by the Snort IDS (Roesch, 2014) and a United Thread Management firewall respectively are used. The configurations also use various parameters including the number of attacking and target hosts for evaluation. For 100 targeted ports, the sequential and parallel distributed portscans are used with multiple scanners. For the sequential portscan, a scanner is selected to scan ports, and when detected by the IDS, another scanner is then selected to continue the portscan. For parallel portscan, the targets and ports are distributed among scanners whose steps are then synchronized. The distributed mechanism works successfully to scan ports despite the presence of firewalls. The experimental results show that for 64 targets, the parallel portscan achieves an average success rate of 84% and continues to perform better than sequential portscan even for small number of targets.

An ARP spoofing based attack (Wu et al., 2010) allows to impersonate ARP messages on the network. It results in providing access of target VMs' packets to an attacker VM. The attacker VM can access private data as well as perform malicious activities. To cope with this attack, a framework as the countermeasure to control communication among virtual machines on a cloud platform is also described. The framework addresses the issue of sniffing and spoofing virtual networks linking VMs. It incorporates multiple layers for routing, sharing and securing the network. The routing layer uses unique tags to monitor the packets, whereas the shared network layer requires VMs of the same organization to share the network. The firewall layer restricts communication of a VM to the outside world and also blocks packets which attempt to modify the routing table.

A technique for detecting botnets in cloud is described in Lin and Lee (2012). The approach initially determines the cryptographic keys being used for botnet communication between bots and Command & Control servers. The attack traffic is decrypted by first identifying patterns of regions that may contain the keys. An

Table 1
Comparison of *network* based attacks on clouds.

Attack Ref.	Attack category	Mechanism	Implications	Vulnerable Components	Countermeasures
McMillan (2009)	A_{1b}	Botnet using Amazon cloud as command and control server	Violation of data protection (*IaaS*)	Cloud network	Lin and Lee (2012), Kourai et al. (2012), Wu et al. (2010), Scarfone and Mell (2007)
Wu et al. (2010)	A_{1c}	ARP spoofing by VM	Violation of data protection (*IaaS*)	Virtual network	Wu et al. (2010), Scarfone and Mell (2007)
Riquet et al. (2012)	A_{1a}	Distributed port scanning	Denial-of-Service (*IaaS* and *SaaS*)	Cloud network	Scarfone and Mell (2007), Karnwal et al. (2012)

Table 2
Comparison of *VM* based attacks on clouds.

Attack Ref.	Attack category	Mechanism	Implications	Vulnerable Components	Countermeasures
Morsy et al. (2010)	A_{2b}	VM creation	Violation of data protection (*IaaS* and *SaaS*)	VM image	Wei et al. (2009), Reimer et al. (2008), Fernandez et al. (2013), Hashizume et al. (2011)
Tandon et al. (2014)	A_{2a}	Cross-VM Cache based side channel attacks	Violation of data protection (*IaaS*)	Shared caches	Liu et al. (2014), Hashizume et al. (2011), Ranjith et al. (2012), Su (2013), Tandon et al. (2014)
Rong et al. (2013)	A_{2d}	Timed scheduling using hypervisor	Theft-of-Service (*SaaS*)	VM Scheduler	Zhou et al. (2011), Wang and Jiang (2010), Murray et al. (2008)
Grobauer et al. (2011)	A_{2b}	VM replication	Violation of data protection (*IaaS* and *SaaS*)	VM image	Wei et al. (2009), Reimer et al. (2008), Fernandez et al. (2013)
Zhang et al. (2012)	A_{2a}	Cross-VM Cache based side channel attacks	Violation of data protection (*IaaS*)	Shared caches	Liu et al. (2014), Hashizume et al. (2011), Ranjith et al. (2012), Su (2013), Tandon et al. (2014)
Rocha and Correia (2011)	A_{2d}, A_{2a}	VM image access and relocation with insecure hypervisor	Violation of data protection, Malicious manipulation of data (*IaaS*)	VM image, Hypervisor	Stolfo et al. (2012), Murray et al. (2008), Rueda et al. (2008), Dawoud et al. (2010), Owens (2009), Zhou et al. (2011), Wang and Jiang (2010), Murray et al. (2008)
Zhou et al. (2011)	A_{2d}	Timed scheduling using hypervisor	Theft-of-Service (*SaaS*)	VM Scheduler	Zhou et al. (2011), Wang and Jiang (2010), Murray et al. (2008)
Garfinkel and Rosenblum (2005)	A_{2b}	VM creation	Violation of data protection (*IaaS* and *SaaS*)	VM image	Wei et al. (2009), Reimer et al. (2008), Fernandez et al. (2013), Hashizume et al. (2011)
Ristenpart et al. (2009)	A_{2a}	VM side channel attack	Violation of data protection (IaaS)	Time shared caches	Liu et al. (2014), Hashizume et al. (2011), Ranjith et al. (2012), Su (2013)
Oberheide et al. (2008b)	A_{2c}	Communication for VM migration and memory access	Violation of data protection, Denial-of-Service (*IaaS*, *PaaS* and *SaaS*)	Hypervisor and network	Xiaopeng et al. (2010), Zhang et al. (2008), Szefer and Lee (2012)
Jasti et al. (2010)	A_{2d}	VM escape and VM hopping to access information of other VMs and impact hypervisor execution	Violation of data protection, Denial-of-Service (*IaaS*, *PaaS* and *SaaS*)	VM and hypervisor	Wei et al. (2009), Szefer and Lee (2012), Hashizume et al. (2011), Su (2013), Liu et al. (2014), Wang and Jiang (2010)
Hlavacs et al. (2011)	A_{2d}	Energy consumption logs to detect VMs being hosted	Violation of data protection (*IaaS*)	VM and storage	Wei et al. (2009), Szefer and Lee (2012), Hashizume et al. (2011), Su (2013), Stolfo et al. (2012)

Table 3
Comparison of *storage* based attacks on clouds.

Attack Ref.	Attack category	Mechanism	Implications	Vulnerable Components	Countermeasures
Harnik et al. (2010)	A_{3b}	Tracing of files/contents through data deduplication and communication covert channel	Violation of data protection, Denial-of-Service (*PaaS* and *SaaS*)	Cloud storage and network	Kaaniche and Laurent (2014), Wu et al. (2014), Somani et al. (2010b)
Grobauer et al. (2011)	A_{3a}	Data scavenging	Violation of data protection (*IaaS* and *SaaS*)	Cloud storage	Wei et al. (2009), Reimer et al. (2008), Fernandez et al. (2013), Leandro et al. (2012), Chadwick (2009), Sanchez et al. (2012)

Table 4
Comparison of *application* based attacks on clouds.

Attack Ref.	Attack category	Mechanism	Implications	Vulnerable Components	Countermeasures
Owens (2010)	A_{4a}, A_{4b}	Fine-grained access for application development, Storage access for shared architectures	Violation of data protection, Denial-of-Service (*IaaS*, *PaaS* and *SaaS*)	Insecure APIs and shared storage	Oberheide et al. (2008a), Liu and Chen (2010), Martignoni et al. (2009), Liu et al. (2011)
Mazurczyk and Szczypiorski (2011)	A_{4a}	Steganography attack through network for data and malicious code	Malicious manipulation of data, Denial-of-Service (*IaaS* and *SaaS*)	Cloud network and storage	Liu et al. (2011), OWASP (2015a), Martignoni et al. (2009), Oberheide et al. (2008a), Liu and Chen (2010), Scarfone and Mell (2007)
Grobauer et al. (2011)	A_{4c}	Protocols vulnerabilities	Violation of data protection (*IaaS* and *SaaS*)	Network Protocols	Gruschka and Iacono (2009), Scarfone and Mell (2007)
Gruschka and Iacono (2009)	A_{4c}	SOAP message manipulation	Denial-of-Service, Theft-of-Service (*PaaS* and *SaaS*)	Web Services and protocols	Gruschka and Iacono (2009), Scarfone and Mell (2007)
Zhang et al. (2014)	A_{4b}	Shared cache based side channel attack	Violation of data protection (*PaaS*)	Shared caches	Doychev et al. (2013), Coppens et al. (2009), Zhang et al. (2014)

entropy search is then performed to identify the keys. Subsequently, the communication taking place between the botmaster and the victims through stepping-stone machines is also captured. It is accomplished by spreading *Pebbleware*, an executable code that identifies the host machines. After reaching the botmaster, it reveals the IP address of the botmaster. The technique of tracing the botnet server is shown to work successfully for the well-known Zeus botnet.

For protecting clouds against stepping-stone attacks, a VM introspection based mechanism is provided in Kourai et al. (2012). The mechanism suggests the use of a packet filter architecture called *xFilter*, which runs in a VMM. The *xFilter* code obtains information regarding the process IDs and user IDs from guest operating systems. Upon intercepting a packet, *xFilter* matches the sender and receiver user IDs with its filtering rules in order to accept or reject the packet. The filtering rules are dynamically updated upon detecting new attacks. A decision cache is also incorporated in *xFilter* which makes it possible to reuse previous decisions thereby reducing the overhead of VM introspection. The suggested filtering approach works with almost 13% of performance degradation of web based communication.

The instrusion detection systems (Scarfone and Mell, 2007) are able to secure a cloud network by analyzing the packets flowing through the network. In contrast to firewalls, the intrusion detection systems analyze traffic pattern through payload information. The intrusion detection systems may provide traffic monitoring at individual host or network level and alert the administrators regarding suspicious activities. Similarly, the intrusion prevention systems are able to discard packets based on the traffic pattern in addition to analyzing packets. A detailed categorization of these systems is given in Section 4.

The research work in Karnwal et al. (2012) targets distributed denial-of-service (DDoS) attacks. The approach initially matches the IP address of the client with already stored IPs. Subsequently, a cloud defender is incorporated to identify suspicious messages and restrict access to avoid DDoS attacks. The cloud defender counts the numbers of requests corresponding to a single IP and filters if a large number of requests arrive from the same IP. It then matches the hop-count value and IP frequency of similar request messages and marks them suspicious. Similarly, the HTTP DDoS attack is handled by using client puzzles which is a part of a WSDL file. The solution of the puzzle is embedded within the header of the SOAP message. The puzzle is sent back to the IP address, and if the puzzle is not resolved by the client machine, the message is discarded. Moreover, a signature is generated while keeping a few parameters twice in the signature, and later added to the SOAP header for XML protection.

3.2. VM based attacks and countermeasures

The security of a virtual machine being copied to create another VM may be compromised by modifying its executable code (Garfinkel and Rosenblum, 2005). The worms/viruses may be injected in the original VM before creating a copy of the VM (Morsy et al., 2010). A compromised virtual machine which also contains the state of the guest operating system thus results in exposing the newly created VM to security threats. Moreover, it is difficult to trace the origin of the vulnerability for the newly created VM.

A cross-VM side channel attack may cause a malicious VM to extract Advanced Encryption Standard (AES) encryption key from a target VM (Tandon et al., 2014). The attack exploits the shared cache by analyzing its access pattern and cache indices when the victim VM executes the AES algorithm.

In Rocha and Correia (2011), various approaches of stealing confidential data in the cloud are described. The approach assumes a malicious insider who has administrative access to VM management. For obtaining passwords, the attack incorporates the access to the memory image of the VM machine using the *xm* command. The passwords can then be retrieved from the memory image by using Linux based utilities. Similarly, to obtain private keys the tool *rsakeyfind* can be used which finds out the key from the memory image of the target VM. For extracting confidential information, the attacker creates a logical disk volume, searches for all the existing volumes, and can mount them to copy the information or files. Another attack that relocates virtual machine can be performed on a cloud. To accomplish this attack, the malicious user first ensures that an integrity-protected hypervisor (Vasudevan et al., 2010) is executing. The integrity-protected hypervisor uses special parameters which make it secure against any possible modifications. A verification process is performed through the key certificates obtained from the TPM of a secure server to make it seem as if the configuration of an insecure hypervisor based machine is the same. The VM is then relocated to the machine with insecure hypervisor thereby making it possible to steal confidential data.

The research work in Zhou et al. (2011) describes a vulnerability in the scheduler of the Xen hypervisor. As the cloud systems use virtualization, the tasks are executed on virtual machines which contain virtual CPUs. The main objective of the scheduler is to determine the mapping between virtual and physical CPUs. Each VCPU is given credits that are debited after scheduling the VCPU. When a virtual CPU goes to sleep state due to I/O call, it retains its credits which cause it to enter the BOOST state on waking up and subsequently preempt other virtual CPUs. To exploit the vulnerability, the scheduler is set to schedule the attacker VM after every 10 ms, let the VM run for a small instance, and go idle subsequently. It ensures that other VM runs for a small time whose credits are therefore reduced. Since the attacker VM wakes in BOOST state, the running VM is preempted and the attacker VM resumes execution, however, the credit balance of the attacker VM never decreases. The experiments performed on an Intel based processor show that the attacker VM can utilize almost 98% of a CPU core's cycles, and on the Amazon EC2 based setup, the attack is able to utilize 85% of the CPU resources. Similarly, a *Time-Stealer* attack (Rong et al., 2013) is described for a recent version of Xen scheduler. The cycle stealing approach works by analyzing the source code. The suggested approach is shown to successfully acquire 96.6% CPU cycles independent of the number of virtual machines executing on the same processor.

While replicating a virtual machine, the private data together with cryptographic keys may be exposed (Grobauer et al., 2011). The private data and keys are supposed to be private to a particular host. However, with a new copy of VM, the exposed data may become public. The data leakage due to VM replication may then be used for malicious activities.

An approach of stealing private information on the cloud due to sharing of physical resources is given in Ristenpart et al. (2009). The approach works in different steps and is based on the fact that VMs for different cloud customers may be executed on the same server. The attacker VM can use side-channel attacks to violate data protection of the target VM. To accomplish that, the malicious user initially attempts to be co-resident on the same system as that of the target VM by probing the network while targeting the port numbers 80 and 443. The web servers are then traced using DNS based probing. When the exposed parameters are used for launching a new instance, it becomes co-resident with the victim VM. The time shared caches are then used to detect the workload of instances of target VMs and launch other attacks. For example, the cryptographic keys can be extracted by using side-channel attacks. Moreover, by finding the keystroke timings, the passwords being entered by the target users can also be recovered.

Various classes of live VM migration attacks are described in

Oberheide et al. (2008b). The control plane class attacks target communication for initiating and managing the overall VM migration process. The data plane class attacks may target the data for leakage of private information. Similarly, the migration module class attacks may gain illegitimate control of the VMM and the guest operating systems. The suggested framework *Xensploit* makes use of the *fragroute* framework to exploit various vulnerabilities such as modifying memory page of a process during transmission, manipulating keys for *sshd* authentication and stack/integer overflow issues.

An attack to access cryptographic keys using the concept of side-channel attack is given in Zhang et al. (2012). The attack works for two separate DomU VMs by assuming that the attacker can access a copy of the software executing on the target VM. The time taken by cache sets of instruction caches is determined. After measuring timings, cache patterns are classified and possible errors called noise are subsequently reduced. Different mathematical operations of a cryptographic algorithm are determined through pattern classification. These sequences of operations are re-constructed to extract the cryptographic key from a victim's machine. The experimentation shows a successful implementation of extracting ElGamal (2006) decryption key from the victim's machine.

Different types of attacks in multi-tenant clouds are discussed in Jasti et al. (2010). The first attack VM hopping may occur when two VMs are set to execute on a single host machine. An attacker on a virtual machine can monitor the traffic of other VMs and subsequently modify their configuration for any malicious activities. The VM Escape attack causes the attacker to access the hypervisor which in turn may be used to affect other running VMs. The attack monitors the CPU and memory utilization and can even cause the hypervisor or VMs to stop executing.

The energy consumption logs may be used as a side channel to recognize the VMs being hosted on a cloud platform (Hlavacs et al., 2011). The energy consumption traces are first sampled followed by computation of probability function for combination of VMs. A statistical analysis is then used to determine the likelihood of VM states.

An approach for improvement of security of the Xen hypervisor is suggested in Murray et al. (2008). The trusted computing base (TCB) of the Xen hypervisor contains the VMM, a privileged virtual machine *Dom0* and other tools which may in turn be used to create other VMs. Since the tools may contain user software, the size of the TCB may run out of bounds. Moreover, the administrator can execute any privileged code which may even impact the functionality of the Xen hypervisor. The suggested approach removes the *Dom0* user space and keeps only the *Dom0* kernel in the TCB. Consequently, the size of the TCB is reduced and the security and integrity of the Xen hypervisor improves significantly.

An approach called *hypersafe* to secure hypervisors with control flow integrity is given in Wang and Jiang (2010). The proposed approach performs memory lockdown to secure code and data. It ensures the pages to be unlocked for modifying even if it is required by the hypervisor itself. Any other code brought for execution in hypervisor space is rejected. To protect control data, a technique called restricted pointer indexing is proposed. It uses control-flow graph to determine the flow of control and generate pointer indexes from control data. The targets in the control flow are found and their access is restricted to pointer indexes. The approach works efficiently as it incurs less than 5% of overhead for protection of code and data.

For tackling various VM based attacks, the misuse patterns are proposed in Hashizume et al. (2011). The misuse patterns describe the environment, conditions and sequences of cloud based attacks including those caused by co-residence of virtual machines and manipulation of virtual machine images. The misuse patterns act as a repository which may then be used by developers for security measures against the attacks.

Two strategies for avoiding cache based side channel attacks targeted at extracting AES keys are described in Tandon et al. (2014). The AES encryption performs various mathematical operations which are very expensive. Consequently, table lookups are used as a replacement for these operations. An attacker can analyze various patterns of the cache to reveal the indices of lookup table which have not been accessed. The attacker detects the execution of the AES encryption as there is a large number of clock cycles. It finds the affected cache sets and performs brute-force method to extract encryption key. The attack can be avoided if the cache is flushed before applying AES algorithm or the lookup table access is made random.

An approach of reducing the possibility of cross-VM side channel attacks using a modified version of scheduler is given in Liu et al. (2014). The suggested approach adds new parameters corresponding to threshold for overlapping of two VMs and noise to be inserted. The threshold for overlapping of two VMs can be adjusted so that the VMs may overlap execution within the threshold limit. As the VMs execution overlapping time approaches the threshold limit, a noise is injected by the scheduler to interrupt the transmission through the side channel. The prototype implementation is shown to successfully mitigate the cross-VM side channel attacks with a very small performance overhead. Similarly, in Su (2013), the concept of a VM Police is proposed to prevent side channel attacks. The Police VM is launched by a host and contains software components as anti-attack units. The scheduling of Police VMs is controlled through several parameters such as load, security and performance requirements.

A mechanism to avoid data leakage using network covert channels is given in Ranjith et al. (2012). A timing covert channel uses delays during communication for encoding and decoding information. Similarly, a network covert channel works when a user types some information in a VM. A keylogger then logs the information and leaks it through the network covert channel to attacker. The solution of these covert channels is to define customized rules for communication between VMs which can be implemented by incorporating a firewall based on VMM. Another covert channel uses the table mapping that corresponds to physical and virtual machine frames. The mapping may be modified to communicate with other virtual machines. It can be restricted by dividing the table so that only the corresponding virtual machine can access it.

To secure data theft attack from a malicious insider, an analysis to detect access patterns is suggested in Stolfo et al. (2012). The approach detects anomalies in data access patterns. For an unauthorized access, the user is returned with decoy information which may not be identified by any user other than owner of the data.

An architecture with flexible enforcement of security policies is described in Rueda et al. (2008). The architecture proposes a configuration phase and a policy enforcement phase. The configuration phase loads a new VM for a specific web application and the enforcement phase ensures authorization of requests from the VMs for end-to-end access control. The architecture establishes secure channels at multiple layers using the given configurations. For a web application, the security policies are downloaded and the browser VM is updated before being loaded. On a URL request, a security label is generated which is then forwarded to the VMM for authorizing the communication as per system security policy and sending it to web server for processing.

For *IaaS* environments, a security infrastructure is proposed in Dawoud et al. (2010). The proposed model contains a secure configuration policy (SCP) to ensure a secure configuration of *IaaS* components. Another component called Secure Resource Management Policy (SRMP) manages the access rules for *IaaS* resources. The third component Security Policy Monitoring & Auditing (SPMA) is able to monitor and track the entire system life cycle. Moreover, the restriction level can be varied depending upon the service provider and requirements.

A cloud security architecture for server virtual machines using policy and threat management services is given in Owens (2009). This architecture is a part of Savvi's general security architecture. The policy management contains various elements including the identity management, Single-Sign-On (SSO), security configuration, vulnerability management and reporting. Each VM is assigned a unique identity which is then used to monitor for any vulnerabilities throughout the VM lifecycle.

The framework *Mirage* proposed in Wei et al. (2009) includes a mechanism for sharing VM images in a secure manner. It contains filters to remove private information or malicious code from the VM image and also contains a mechanism for tracking operations applied to the VM image. After publishing the image, the framework may also be used for scanning and fixing of viruses or malicious software. Similarly, a special VM image storage format (Reimer et al., 2008) is proposed to secure images by exploiting semantic information in the images. It makes use of a manifest corresponding to an image and a store to contain image data which may be converted to the original image form. In Fernandez et al. (2013), a repository is incorporated to secure the VM images in a cloud. It uses a monitor to scan images, an authenticator to authenticate the legal users and an auditor to track access to image.

A framework for providing life-cycle protection of VM and virtual network called *VNSS* is provided in Xiaopeng et al. (2010). The framework contains a controller and multiple agents as components for securing virtual computing environments. The controller component loads VM configuration and calls an agent to create an instance of the virtual machine. Another agent for generating security policies is then invoked by the controller. Similarly, for VM migration, the controller uses the VM migration agent together with the agents for security context migration and security policy migration. The VM destruction phase is carried out by the controller by destroying the VM instance and removing the security policies. The experimental results show a successful lifecycle implementation with uninterrupted execution and secure migration for FTP applications.

For secure migration of VMs, a framework called *PALM* is described in Zhang et al. (2008). The framework works for Xen VMM and makes use of different modules for protection during migration. The data protection module performs encryption and decryption of data related to protected processes. The migration manager controls the overall migration process, whereas other modules manage the transmission of metadata and protect it from various security vulnerabilities. With the *PALM* framework, there is a small performance degradation in comparison with the live migration by Xen hypervisor.

A virtual machine executing on a cloud may be temporarily suspended for maintenance activities and resumed later on. While suspending it is possible to save the current state of disk, memory and CPU. This feature may be used to perform malicious activities as it becomes possible to perform multiple attempts for login, and then rollback the VM to its previous state. The countermeasures for these attacks work by using the log of VM activities or enforcing the features of suspend/resume operations to work with user input (Szefer and Lee, 2012).

3.3. Storage based attacks and countermeasures

Storage services of a cloud make use of data deduplication in order to keep a single copy of data. In Harnik et al. (2010), different attacks related to data deduplication in clouds providing storage services are described. The first attack makes it possible for the attacker to identify the uploaded files. The second attack allows to determine the contents of a file stored on a cloud server. Similarly, the third attack allows to create a covert channel for malicious activities. The channel can make communication possible between a malicious software and a control server.

With elastic clouds, the data scavenging may occur when the resources allocated to a user are re-allocated to another user (Grobauer et al., 2011). Despite a new allocation, the data and storage of previous user may become accessible to the new user. The violation of data protection may therefore occur due to data scavenging.

A mechanism for detecting covert channels in clouds using a framework called *C^2Detector* is provided in Wu et al. (2014). A covert channel can be used to violate data confidentiality provided by a cloud platform. To cope with the covert channels of CPU load, memory and cache based, an automaton having four states is incorporated. To detect the covert channels, change pattern of shared resources is matched with the Markov model. A match of the pattern implies the transfer of confidential information through the covert channel. Similarly, for operation sequences, Bayesian model is incorporated, which takes input from the Markov detector. A deviation from the Bayesian model implies a covert channel to exist in the cloud. The *C^2Detector* successfully detects covert channels with a small number of false positives.

For securing data during transmission, a mechanism using digitized signatures is described in Somani et al. (2010b). The suggested approach performs digital signatures using RSA. A hash function is initially applied to generate a message digest which is subsequently encrypted. The encrypted text can then by decrypted for verification. Similarly, the approach given in Kaaniche and Laurent (2014) proposes a mechanism to share data on a public cloud using deduplication. The approach works in a secure manner by encrypting data and encapsulating rights in a separate file. The decryption of data can be performed only by authorized users.

The VM image access control mechanisms (Wei et al., 2009; Reimer et al., 2008; Fernandez et al., 2013) discussed in the previous section may also be applied to secure data scavenging.

With multiple organizations sharing user information, a federated identity management system can be incorporated. Leandro et al. (2012) present a federated identity based authorization scheme which deploys the Shibboleth authentication and authorization system (Chadwick, 2009) for management of identities. It supports Single-Sign-On (SSO) mechanism so that multiple organizations within a federation may share identity information. Consequently, a user is not required to login repeatedly to access resources multiple times. Similarly, Sanchez et al. (2012) describe an identity management system. Their approach employs the Security Assertion Markup Language (SAML) (Ragouzis et al., 2008) adapted for dynamic cloud federation. The SAML language uses XML for communicating data required for authentication and authorization between organizations. The proposed model is then used to provide access to cloud resources while retaining the user privacy.

3.4. Application based attacks and countermeasures

Various security challenges faced due to elasticity of clouds are described in Owens (2010). A main challenge is provision for a fine-grained access in a cloud environment. Moreover, an elastic model has to cope with the time and context parameters for restraining various actions. Similarly, the data on a shared architecture may not be secure due to environment access to other users.

For steganography attacks, secret data called steganogram can be embedded within normal data exchange which may not be detected by third parties. The secret data may contain malware code which may result in security breach (Mazurczyk and Szczypiorski, 2011).

The data protection may be violated due to the protocol vulnerabilities (Grobauer et al., 2011). The protocols being used for access to cloud data and services are described as an insecure entity which may become potential threats for cloud computing.

For access to cloud infrastructure, Web Services are used by cloud providers. The vulnerability of these services for cloud infrastructure is exposed in Gruschka and Iacono (2009). The services make use of Simple Object Access Protocol (SOAP) messages for cloud requests. A SOAP message can be manipulated by intercepting it and modifying the request. The SOAP message header contains signatures which are kept unchanged together with *Body wsu:Id*, while the message body is replaced with a new bogus request. An invalid request validated by the cloud provider makes the entire cloud vulnerable. This attack can be restricted by validating XML schema which ensures that multiple occurrences with the same *Id* do not occur. Moreover, the security policy validation mechanism may be improved to counter such attacks.

For *PaaS* environments, a framework to perform side-channel cache-based attacks is given in Zhang et al. (2014). The control flow graph of an executable is used to detect memory chunks that are monitored for the attack. Subsequently, an NFA with states corresponding to memory chunks is constructed. The execution of a victim's application is traced and a malicious instance of the attacker code is co-located with the application. The malicious code can then extract information from the application by making use of cross-site requests. The suggested approach is shown to successfully reveal information regarding a user's shopping items on a victim website, resetting passwords of users on a website and breaking XML encryption.

For securing cloud computing platforms, the antivirus software may be of great significance as it may monitor and hinder any malicious code to impact the cloud. In Oberheide et al. (2008a), the architecture of an antivirus CloudAV is described. The antivirus supports in-cloud detection of malicious code by incorporating multiple detection engines. These engines execute in parallel and can analyze the files being transferred. A host agent runs on host systems to send suspicious files to network for analysis. The network service analyzes the code to identify threats. A forensic analysis is also used to track information related to file access and extract the actions performed by malicious code. Using multiple detection engines, detection coverage of the CloudAV antivirus increases significantly and achieves almost 94% of average detection rate.

A mechanism incorporating the detection of malware using historical information is described in Liu and Chen (2010). The approach makes use of logs corresponding to writing or creation of executable files in the Portable Executable (PE) format. With a lightweight log collector, any changes made to PE files by malware code are captured. The logs are processed by using MapReduce which performs file indexing and relation indexing, to represent the locations and relationships of PE files respectively. The experimental results show 83% accuracy for detecting malware.

A framework for dynamic analysis of malware and suspicious code is given in Martignoni et al. (2009). The approach effectively utilizes the resources of the cloud as well as the end users or clients who may be victims of malware. It causes the clients to send malware programs to a cloud for analysis on its behalf. The suspicious code is executed on the cloud except the environment independent system calls which are executed on the users' systems, and their output is then submitted to the cloud. The output of the code is analyzed thoroughly to determine whether there is any malicious activity performed by the suspicious code. This approach works efficiently to detect malware as it mitigates the overhead of executing a large part of the code on the end users' system and transfers it to a resource rich cloud.

For steganography attacks in cloud storage systems, a StegAD scheme is described in Liu et al. (2011). The scheme incorporates two algorithms for detecting the affected files and determine the hidden places respectively. The scheme is shown to successfully work for audio files based attacks on cloud storage. Similarly, for securing web applications and thwarting malware injections through networks, the application firewalls (OWASP, 2015a) may be incorporated together with intrusion detection systems (Scarfone and Mell, 2007).

The timing information leakage using shared resource based attacks may be tackled through a transformation called *if-conversion* (Coppens et al., 2009). The suggested approach works for control-flow based side channels in which program execution is traced by finding the path followed during execution of the program. The transformation of code for conditional execution protects against the control flow based side channel attacks. Likewise, a framework to analyze cache based side channel attacks is described in Doychev et al. (2013). A binary file and cache parameters are input to the framework for analysis and detection of attacks. The framework uses the parser and iterator components together with abstract domains to analyze the attacks.

4. Automated cloud protection using intrusion detection and prevention systems

An Intrusion Detection System (IDS) analyzes the packet header and payload to compare it with any anomalies found in comparison with the normal traffic. This is in contrast to a firewall which filters the network traffic by examining the packet headers flowing through the network ports. For anomalous traffic, an IDS attempts to identify the pattern against common threats, and alerts the network administrator. An Intrusion Prevention System (IPS) works just like an IDS, however it may also reject the packets or terminate the connection. Since the backbone of a cloud based platform is usually a high-speed network, it must be protected by a fully automated intrusion detection/prevention system. A network intrusion detection/prevention system (NIDS/NIPS) attempts to secure all computer systems in a network, whereas a host-based intrusion detection/prevention system (HIDS/HIPS) attempts to secure a single host. A highly *scalable* intrusion detection system is able to provide support for efficient utilization of modern high performance architectures.

Two different types of detection models have been incorporated in intrusion detection systems: Statistical or Signature-based (Scarfone and Mell, 2007). The statistical model maintains profiles regarding users, hosts, applications and connection (ports, devices and protocols). It then compares current activity with the attributes of the profile for any anomalies. In contrast, a signature-based model compares the traffic against a collection of signatures or threat patterns.

Table 5 provides a comparative analysis of the main intrusion detection/prevention systems in terms of category, intrusion detection model, main components, technical complexity, scalability and open source support parameters.

4.1. ACARM-ng

The *Alert Correlation, Assessment and Reaction Module-next generation* system (Balcerek et al., 2012; WCSS, 2010) is able to generate alerts and correlate them. For intrusion detection, it can work for networks as well as for individual hosts. It gets input in Intrusion Detection Message Exchange Format (IDMEF) (Debar et al., 2007), a standard format for communicating with IDS. Its

Table 5
Comparison of major intrusion detection/prevention systems.

Framework	Category	Detection model	Major components	Technical complexity	Scalability	Open source
ACARM-ng (Balcerek et al., 2012; WCSS, 2010)	IDS, IPS	Profile-based model to support NIDS, NIPS, HIDS and HIPS	Correlation daemon, database engine	Medium	Low	Yes
Suricata (OISF, 2015)	IDS, IPS	Signature-based model to support NIDS and NIPS	Network Intrusion detection, prevention and monitoring engines, analyzer and logger	Medium	High	Yes
OSSEC (Hay et al., 2008; OSSEC, 2015)	IDS, IPS	Signature-based model and profile-based model to support HIDS and HIPS	File integrity and log checkers	High	Low	Yes
Snort (Roesch, 2014)	IDS, IPS	Signature-based model and profile-based model to support NIDS and NIPS	Packet sniffer, pre-processor, detection engine and logger	Medium	Medium	Yes
NIDES (Anderson et al., 1995)	IDS	Signature-based and profile-based model to support NIDS	Analyzer, resolver and archiver	High	Low	No
eXpert-BSM (Lindqvist and Porras, 2001)	IDS	Profile-based model to support NIDS	Inference engine, knowledge base and analyzer	High	Low	No
Fail2ban (Jaquier, 2013)	IDS, IPS	Profile-based model to support NIDS and NIPS	Client, Server and *Jails*	Low	High	Yes
Prelude-OSS (CS, 2015)	IDS	Profile-based model to support NIDS	Manager, correlator and log file manager	Low	Low	Yes
Sagan (2015)	IDS	Profile-based model to support NIDS	Event and log analyzer	Medium	High	Yes
Samhain (2006)	IDS	Profile-based model to support HIDS	Integrity checker, log server and console	Medium	High	Yes
Bro-IDS (Paxson, 1999)	IDS, IPS	Profile-based model to support NIDS	Script interpreter and event engine	High	Low	Yes

architecture makes use of filters (for joining and correlating attacks), triggers (for reactions to attacks) and a database (for storing alerts and state of the application).

4.2. Suricata

The *Suricata* intrusion prevention system (OISF, 2015) is a rule based engine that supports intrusion detection and prevention by monitoring network traffic. It generates alerts for the system administrator if any suspicious activity is performed in the network. On high performance architectures, it can scale efficiently by exploiting multi-threading feature on multiple processors or cores. Its detection system can identify protocols thereby making it easy for users to obtain security based on protocols instead of ports.

4.3. OSSEC

The *Open Source SECurity* intrusion prevention system (Hay et al., 2008; OSSEC, 2015) monitors the network for individual hosts. Its implementation uses several daemons to perform specialized tasks such as forwarding logs to servers, analyzing logs and generating alerts. It uses a central manager (to retrieve information from *syslog*), agents and databases to generate alerts on a real-time basis. The manager stores the rules and the configuration parameters together with the system logs and events.

4.4. Snort

Snort (Roesch, 2014) is a network intrusion prevention system which provides support of packet sniffing and packet logging modes in addition to analysis in NIDS mode. Through packet sniffing, the packet stream is captured and displayed to the user, whereas through packet logging, the packets may be stored on a backup storage device. To generate alerts, it uses the components of packet sniffer, pre-processor, detection engine and logger. The packet logging may speed up using *fast* mode which makes the logger write logs efficiently in binary form.

4.5. NIDES

The *Next Generation Intrusion Detection Expert System (NIDES)* (Anderson et al., 1995) executes on a host to analyze user activities on a collection of target computers. It is able to perform real-time monitoring for suspicious behaviors. It uses analyzers for statistical and rule-based analyses to detect unusual behavior and match with known intrusion types. Subsequently, the alerts are reported to relevant users. The NIDES system also includes a resolver to filter the alarms for security officer and an archiver to store alerts together with analysis results.

4.6. eXpert-BSM

The *eXpert-BSM* software (Lindqvist and Porras, 2001) is a host-based intrusion detection system which uses a knowledge base to detect intrusions and generate alarms on a Sun-Solaris based system. It performs an analysis of audit trails by making use of inference engine and knowledge base rules. Its expert system can monitor the system for suspicious network activities and traffic flowing through the user-configured ports.

4.7. Fail2ban

The *Fail2ban* software (Jaquier, 2013) uses log files and detects patterns which may correspond to intrusion efforts. After identifying the break-in attempts, it may add rule to firewall and generate an alarm for the system administrator. It uses a server component to listen to ports, a client component to send commands to server and a *jail* component to represent combination of filters and actions. Corresponding to a filter, one or more actions such as adding firewall rule, getting information of attacker and sending alert to administrator may be performed.

4.8. Prelude-OSS

The *Prelude-OSS* intrusion detection system (CS, 2015) is an open source version of the Prelude software to provide support of security event management. The open source version works for small organizations to provide the basic functionality of intrusion detection. It makes use of several modules for accessing database, receiving and storing events, correlating alerts and log management. The intrusion events occurring in a network are displayed through a graphical interface.

4.9. Sagan

The *Sagan* software (Sagan, 2015) can perform a real-time analysis of intrusion events. For better scalability, it has a multi-threaded architecture for event and log analysis. It provides support of detecting and managing intrusion events in a standard manner which makes it compatible for correlating log events with other intrusion detection systems. The events and alerts may also be written to a database. Moreover, it uses diverse output formats for event detection, firewall support and real-time alert management.

4.10. Samhain

The *Samhain* intrusion detection system (Samhain, 2006) is a host-based IDS which may perform file integrity checking, port scanning and log analysis. With a client/server based architecture, it uses the components of integrity checker, log server and a web-based console. It also provides support for central logging and storage in database. Its web-based console allows to view client/server activities and analysis reports.

4.11. Bro-IDS

The *Bro* network analyzing framework (Paxson, 1999) can be used to detect intrusions through real-time monitoring. It contains an interpreter to execute policy scripts, an event engine to manage events and *libpcap* (Tcpdump, 2015) to monitor packet streams. The event engine obtains packet stream and ensures that the packets are well-formed. Similarly, a checksum on IP headers is also performed. An invalid packet is then discarded and an event is triggered to communicate the problem. The interpreter executes the event handlers, generates new event notifications and logs data to disk.

5. Securing cloud execution environment through trusted cloud computing

Trusted cloud computing enables the cloud service providers to ensure a secure and confidential execution environment while maintaining integrity of its data and computations. In this section, we analyze various contributions aimed at securing cloud computing environment through trusted computing. Table 6 provides a parametric comparison of research contributions aimed at trusted computing based security for cloud computing platforms. The comparison is performed in terms of mechanism, major components, cloud layers, automation level and encryption/certificates used in the approach.

For trusted computing, a field-programmable gate array (FPGA) can be deployed to securely identify a computation implemented in the logic fabric (Ken and Ramarathnam, 2012). A symmetric encryption key is stored on FPGA memory. The FPGA can then be installed in a cloud server. A trusted authority in the cloud can encrypt and sign applications with the keys of FPGAs. Consequently, the application can process data in a secure manner.

A trust overlay network suggested in Hwang and Li (2010) uses distributed hash tables to provide support of intrusion detection and prevention to DDoS attacks. A distributed security mechanism is incorporated while making use of cloud resources. A data object in the cloud environment is protected by using data coloring based watermarking. Different security levels are represented by data colors whose characteristics are known only to owners and cannot be detected by cloud providers or other users without having known the characteristics.

A trust management module is deployed for establishing secure communication between cloud users and cloud providers (Takabi et al., 2010). The approach uses a trust integrator for maintaining trust between service providers and between users and service providers. The trust integrator works by discovering service providers, negotiating parameters and generating groups for services. To accomplish the task, the service integrator contains different modules to support security, trust and service management. For diverse policies of service providers and identity management, an ontology based heterogeneity management module is proposed together with a user-centric identity management mechanism.

To secure runtime environment of a cloud, a watermark based approach (Fu et al., 2010) is suggested. Different algorithms of watermark have been incorporated to secure Java programs. Moreover, for execution of the watermarked Java programs, the JVM is also customized to recognize and extract watermarks. For securing watermark recognition, the watermark is concealed in the JVM. Upon a mismatch, the watermark may not be extracted for the Java program being executed on the cloud. Consequently, the program terminates with an error message that is subsequently communicated to the cloud provider to restrict execution of such Java program. The proposed approach results in secure execution of Java programs on cloud platforms.

A mechanism for developing a trusted third-party is described in Zissis and Lekkas (2012). The trusted third-party ensures a secure communication and transactions b/w two parties (Castell, 1993). The suggested solution implements confidentiality through IPSEC and SSL for communication b/w machines. The authentication process includes usage of digital signatures together with Single-Sign-On (SSO) and Lightweight Directory Access Protocol (LDAP) which is used to access information regarding users and resources on a network. Moreover, security domains are created to make federated clouds which represent clouds communicating through standard interfaces. For privacy of data, hybrid cryptography employing both symmetric and asymmetric encryption mechanisms is proposed. Similarly, an attribute based authorization using certificates is proposed to be implemented for a trusted third-party.

An integrity model for management of different parameters of virtual machines is presented in Jansen et al. (2008). The prototype model is implemented using the Xen hypervisor. It uses the concept of Trusted Platform Module (TPM) (Trusted Computing Group, 2011) to work for virtual machines. The module works for the enforcement of security policies and guarantee compliance while attaching new devices to virtual machines. It stores a log of system history in terms of policies and configurations. The attestation and sealing/unsealing mechanisms are then implemented to enforce access restriction. Consequently, the security policy may not be modified for unauthorized usage.

A collaborative model to support trusted cloud computing

Table 6
Comparison of trusted computing based cloud computing security.

Reference	Mechanism	Major Components	Cloud Layer	Automation Level	Encryption/Certificates
Ken and Ramarathnam (2012)	TPM on FPGA	Trusted Authority, TPM and FPGA	*SaaS*	High	RSA, SHA, AES
Shen et al. (2010)	Role-based access using Trusted Computing Platform (TCP) integration with Trusted Platform Support Service (TSS)	Trusted Platform Support Service with TPM	*IaaS*, *PaaS* and *SaaS*	Low	Generic with X.509 certificates
Hwang and Li (2010)	Distributed hash table based overlay networks for protecting objects using data coloring and watermarking	Trust overlay network and hash tables	*IaaS*, *PaaS* and *SaaS*	Low	RSA and watermarking
Takabi et al. (2010)	Discovery of services, Role-based access, trust and identity management	Service integrator having modules for security, trust and service management	*IaaS*, *PaaS* and *SaaS*	Low	Generic
Fu et al. (2010)	Watermarking Java program, Recognition and extraction of watermark by JVM	Watermark embedder, JVM generator and deployment modules	*SaaS*	High	Watermark-based
Santos et al. (2009)	A trusted platform using trusted VMM to securely execute guest VMs and attest the cloud infrastructure providers	TPM, Trusted VMM and Coordinator	*IaaS*	Low	Generic
Jansen et al. (2008)	Security policies for virtual machines using TPM	Compartment, Integrity and Secure virtual device managers	*IaaS*	High	Generic for VM image and Network configuration
Zissis and Lekkas (2012)	Trusted third-party using IPSEC, SSL, SSO and LDAP	Certificate, data, authentication, LDAP and database servers	*IaaS*, *PaaS* and *SaaS*	Low	IPSEC, SSL, SSO and LDAP
Alhamad et al. (2010)	Designing SLA parameters, selecting cloud service provider and monitoring	SLA agent and services directory	*SaaS*	Low	SLA-based
Li et al. (2010)	Multi-tenancy trusted computing environment	Third-party auditor, TPM and SLA	*IaaS*	High	Generic
Yang et al. (2010)	Collaborative trust model using trust domains, trust tables and historical data to compute trust values	Trust domains and trust tables	*IaaS*	Medium	Generic
Manuel et al. (2009)	Trust model using security, feedback and reputation evaluators to find trust values	Resource broker and evaluators	*IaaS*	Medium	Kerberos and PERMIS based authentication

using firewalls is given in Yang et al. (2010). The model works for environments where cloud service providers have diverse policies. The model is organized into nodes and the collection of nodes called *domains*. A trust table is deployed in the model to maintain trust values corresponding to nodes. Upon a user request, the domain agents send the message to the neighborhood agents to communicate with the firewall. A digital signature is required by the cloud service provider for initial connection before allocating resources to users. The trust values are updated dynamically based on the history of transactions.

A trusted cloud computing environment for *IaaS* model using a 2-level hierarchy is given in Li et al. (2010). The environment incorporates a third-party auditor to verify the trustworthiness of cloud service providers. The cloud platform is attested through a policy based attestation model. To check conformance of the cloud service provider, the proposed model requires security parameters to be agreed upon through the service level agreement (SLA).

A trusted computing platform (TCP) can be integrated into a cloud environment for trusted cloud computing (Shen et al., 2010). A TCP is based on TPM and may be used for authentication and role based access. A user logs on to a cloud using TCP and obtains a trusted certificate from the cloud. For secure communication, the client uses the certificate and information regarding the role. Moreover, the data security is ensured using the keys generated with TPM.

For clouds and grids, a trust management system is proposed in Manuel et al. (2009). The suggested model calculates trust values using different evaluators corresponding to security, feedback and reputation parameters. The security evaluator uses the authentication and authorization types for assigning different trust values. The feedback evaluator obtains input from the user to assign trust values. Similarly, the reputation evaluator takes into consideration the capability of cloud or grid based system in terms of its resources. A trust manager then accumulates the trust values which is then communicated to other components for execution of the user request.

A trusted cloud computing model based on SLA is proposed in Alhamad et al. (2010). The proposed architecture uses SLA agent for defining SLA metrics, selecting cloud service providers and monitoring the business activities. The trust management model takes input from the cloud service providers, users and SLA agents to rank the cloud providers for selection. Moreover, a directory of cloud services is suggested to store information regarding cloud providers and their services. The mechanism of trust establishment requires the cloud services to be advertised, followed by the selection of cloud providers. The SLA agreement is communicated to the user requesting the resources. Further communication with the cloud provider takes place if the user agrees to the SLA agreement.

6. Regulating cloud security compliance issues

Various regulatory bodies have defined rules and regulations to ensure security of data and allowing disclosure under permissible circumstances. The rules defined by these regulatory bodies encompass a wide range of applications and practices as detailed below.

6.1. Common criteria compliance

Common criteria (CC-Portal, 2015) has become a global standard for evaluation of security products. It makes use of protection profiles which specify security requirements in an implementation independent manner. It assigns different Evaluation Assurance Levels (EALs) ranging from EAL1 to EAL7 to represent different grades of security assurance. The cloud related products are also evaluated and certified if they comply with the security requirements. For instance, the VMWare ESXi, vCloud Networking & Security (VMWare, 2015), Citrix Xen Server (CESG, 2012), z/VM (IBM-Inc., 2015) and the KVM hypervisor (RedHat-Inc., 2015) all are *Common Criteria* certified.

6.2. Trusted computing compliance

The Trusted Computing Group is a non-profit group of companies formed for promoting trust and security in computing platforms (TCG, 2015a). The group has developed a Trusted Platform Module (TPM) specification to support trusted computing. Using TPM, the cryptographic information is stored on hardware to protect it from software based attacks. The cloud based servers can be easily secured by using a TPM (TCG, 2015b). Moreover, the encryption and authentication standards developed by TCG are implemented for securing data storage and data communication on a cloud platform respectively.

6.3. Privacy acts compliance

Governmental organizations as well as private institutions or individual customers may store their data or public data on clouds. The privacy and confidentiality of data needs to be ensured by cloud service providers due to legal implications as mentioned in different governmental rules and statutes.

6.3.1. Privacy of health related information

The Health Insurance Portability and Accountability Act (HIPAA) (U.D. of Health & Human Services, 2007) describes a set of privacy rules to provide support of confidentiality and security of health related information. In addition to protecting data during transmission, the personal health information must be protected while being maintained in any form. Even it requires the data to be protected against threats and malicious software. Moreover, the data backup and disaster recovery plans need to be implemented. The information disclosure without obtaining a patient's consent is allowed only when mandated by law. Several cloud storage providers including the CareCloud (2015), FireHost (2015), Symform (2014) and Carbonite (2015) claim to be HIPAA compliant by ensuring implementation of its standards.

6.3.2. Privacy of electronic data

The Electronic Communications Privacy Act (ECPA) (DHS, 2013) describes rules and restrictions regarding protection during transmission of electronic data. It encompasses rules for access to private data in stored form as well as interception of data during communication. An unauthorized access is prohibited if data is stored on third-party storage devices, however ECPA allows official access to data without informing the owner thereby reducing the confidence of customers on cloud providers. A coalition called *Digital 4th* is currently working to recommend modification to ECPA for making it more appropriate for the cloud computing platforms (Digital 4th Coalition, 2015).

6.3.3. Privacy of financial data

The Fair Credit Reporting Act (FCRA) (FTC, 2015) describes rules for privacy of user credit information. For a credit reporting agency storing credit data of customers on a cloud, it becomes compulsory to ensure its security through FCRA compliance. Similarly, some safeguard rules are described in the Gramm–Leach–Bliley (GLB) Act (GPO, 2015) for financial bureaus to ensure confidentiality of data. The financial institutions are required to only select the service providers who are able to implement the safeguard rules.

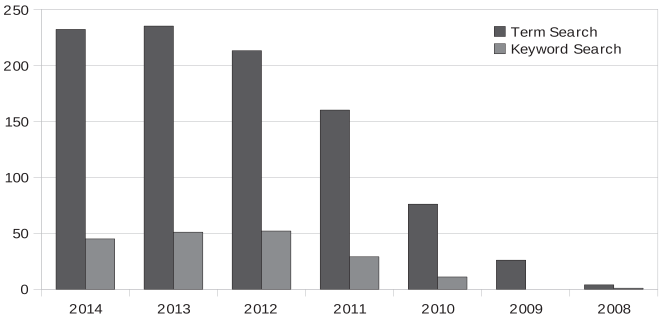

Fig. 3. Cloud security contributions.

7. Cloud security issues in the future

Together with the growth of cloud services, the cloud vulnerability incidents have been on the rise. The IBM statistics (IBM, 2014) show that for an organization on average, the number of cyber security attacks has reached up to 1400 per week. The conventional brute-force, vulnerability scan, web application and malware/botnet attacks have been reported to dominate in year 2014 (AlertLogic, 2014). With the increasing number of attacks, the number of research contributions describing countermeasures to ensure cloud security has also increased significantly. Fig. 3 presents a year-wise trend with the number of research papers citing the term 'cloud security' (Term and Keyword search retrieved from the ACM Digital Library).

Despite the countermeasures being suggested in research, many issues seem to pose challenge for securing cloud in the future.

7.1. *Trusted execution environment*

A cloud service provider must ensure a trusted environment for its clients. The code and data stored on the cloud storage should only be accessible to authentic relevant users. In this regard, trusted execution technologies provide a reliable way to verify the integrity of the system (Futral and Greene, 2013; Intel, 2013). Since the integrity of a system may span hypervisor as well as the applications, management software and security policies, any compromise to their integrity may be identified and necessary measures may be taken to protect the system subsequently.

7.2. *Protocol vulnerabilities*

To access data and services on clouds, the protocols defined for communication have proved to be vulnerable to various attacks. For instance, the SOAP message can be manipulated to target cloud platform services and violate data protection (Karnwal et al., 2012; Gruschka and Iacono, 2009). Similarly, the insecure interfaces and APIs used to interact with cloud systems have been reported to be the top threat (Ko et al., 2013). It is therefore necessary to mitigate the vulnerability of already existing protocols. Consequently, either a secure implementation of these protocols is required or a strong encryption mechanism needs to be incorporated to improve the security (Karnwal et al., 2012; Mearian, 2013).

7.3. *Federated identity interoperability*

The identity and access management provides support of controlling users' access to shared cloud resources through individual user identities. A federated identity management makes multiple organizations share the identities to support Single-Sign-On mechanism (Leandro et al., 2012; Chadwick, 2009; Sanchez et al., 2012). As the federated identity model continues to evolve, the diversity of protocols, conformance requirements and architectures makes it complicated for deployment among multiple organizations (Maler and Reed, 2008). The open source identity stack based implementations (Craig et al., 2014) may be beneficial for addressing interoperability issues.

7.4. *Open standards compliance*

Despite having identified common threats and vulnerabilities in cloud systems, the requirement for the minimum security needs to be leveraged by defining open standards. Although currently various virtualization products comply with the common criteria standards, the compliance of cloud security from different perspectives such as services, deployment and interoperability is still undefined (VMWare, 2015; CESG, 2012; Lewis, 2013).

8. Conclusion

Cloud computing offers services for consumers through effective utilization of shared resources. Despite its effectiveness for cloud service providers as well as for the cloud users, its prevalence is hindered by various security issues. This paper presents a comprehensive survey of the issues and research contributions aiming at cloud security. These issues encompass security of data and services on cloud platforms. We categorize security threats and perform a comparative analysis of security issues and the countermeasures suggested in the literature to cope with these issues.

We also survey the main intrusion detection and prevention systems and analyze their effectiveness in terms of working mechanism, components and scalability. Moreover, a comparative analysis of the contributions made for trusted cloud computing is also presented. We also analyze a large number of standard acts and regulations required for compliance by the cloud service providers. In the perspective of future challenges, we discuss the main issues related to cloud security and their possible solutions in terms of the trusted execution, protocol vulnerabilities, federated identity interoperability and open standards compliance.

References

Balcerek, B., Szurgot, B., Uchroński, M., Waga, W., 2012. ACARM-ng: next generation correlation framework. In: Bubak, M., Szepieniec, T., Wiatr, K. (Eds.), Building a National Distributed e-Infrastructure – PL-Grid: Scientific and Technical Achievements Science, vol. 7136. Springer, Berlin, Heidelberg, pp. 114–127, ISBN: 978-3-642-28267-6.

AlertLogic, 2014. Cloud Security Report: Research on the Evolving State of Cloud Security. URL ⟨https://www.alertlogic.com/resources/cloud-security-report/⟩, May.

Alhamad, M., Dillon, T., Chang, E., 2010. SLA-based trust model for cloud computing. In: 2010 13th International Conference on Network-Based Information Systems (NBiS), pp. 321–324. http://dx.doi.org/10.1109/NBiS.2010.67.

Anderson, D., Frivold, T., Tamaru, A., Valdes, A., 1995. Next Generation Intrusion Detection Expert System (NIDES), Software Users Manual, May.

Ashford, W., 2015. Cyber Criminals Turn their Attention to Cloud Service Credentials. URL ⟨http://www.computerweekly.com/news/2240241940/Cyber-criminals-turn-their-attention-to-cloud-service-credentials⟩.

Ashktorab, V., Taghizadeh, S.R., 2012. Security threats and countermeasures in cloud computing. Int. J. Appl. Innov. Eng. Manag. 1, 234–245.

Aviram, A., Hu, S., Ford, B., Gummadi, R., 2010. Determinating timing channels in compute clouds. In: Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop, CCSW '10. ACM, New York, NY, USA, pp. 103–108.

AWS, 2014. Amazon Web Services: Overview of Security Processes, pp. 1–68.

Bisong, A., Rahman, S.M., 2011. An overview of the security concerns in enterprise cloud computing. Int. J. Netw. Secur. Appl. 3 (1), 30–45.

Buyya, R., Broberg, J., Goscinski, A.M., 2011. Cloud Computing Principles and Paradigms. Wiley Publishing, New Jersey, USA.

Carbonite, 2015. Carbonite Supports HIPAA Compliance. URL ⟨http://www.carbonite.com/online-backup/business/hipaa-compliance-encrypted-backup⟩.

CareCloud, 2015. HIPAA-Compliant Cloud Storage – Cloud Security and Data Control for Medical Enterprises. URL ⟨http://www.carecloud.com/hipaa-compliant-cloud-storage/⟩.

Castell, S., 1993. Code of Practice and Management Guidelines for Trusted Third Party Services, INFOSEC Project Report S2101/02.

CC-Portal, 2015. Common Criteria: Protection Profiles. URL ⟨http://www.commoncriteriaportal.org/rss/pps.xml⟩.

CESG, 2012. Certification Report No. CRP270 – Citrix Xen Server 6.0.2 Platinum Edition. URL ⟨www.commoncriteriaportal.org/files/epfiles/XenApp7-STv1-0.pdf⟩.

Chadwick, D., 2009. Federated identity management. In: Aldini, A., Barthe, G., Gorrieri, R. (Eds.), Foundations of Security Analysis and Design V, Lecture Notes in Computer Science vol. 5705. Springer, Berlin, Heidelberg, pp. 96–120.

Cloud Security Alliance, 2010. Top Threats to Cloud Computing v1.0, pp. 1–14.

Cloud Security Alliance, 2012. Secaas Implementation Guidance, Category 7: Security Information and Event Management, pp. 1–33.

Coppens, B., Verbauwhede, I., De Bosschere, K., De Sutter, B., 2009. Practical mitigations for timing-based side-channel attacks on modern x86 processors. In: 2009 30th IEEE Symposium on Security and Privacy, pp. 45–60.

Craig, M., Frost, L., Jang, M., Egloff, A., 2014. Openidm Release Notes Version 3.1.0. URL ⟨http://docs.forgerock.org/en/openidm/3.1.0/OpenIDM-3.1.0-Release-Notes.pdf⟩.

CS, 2015. Prelude OSS. URL ⟨http://www.prelude-siem.com/index.php/uk/products/87-products/98-prelude-oss-en⟩.

Curran, K., Carlin, S., 2011. Cloud computing security. Int. J. Ambient Comput. Intell. 3 (1), 14–19.

Digital 4th Coalition, 2015. A Summary of the Electronic Communications Privacy Act (ECPA). URL ⟨http://www.digital4th.org/the-problem/⟩.

Dawoud, W., Takouna, I., Meinel, C., 2010. Infrastructure as a service security: challenges and solutions. In: 2010 7th International Conference on Informatics and Systems (INFOS), pp. 1–8.

Debar, H., Curry, D., Feinstein, B., 2007. The Intrusion Detection Message Exchange Format (IDMEF). URL ⟨https://www.ietf.org/rfc/rfc4765.txt⟩.

DHS, 2013. Electronic Communications Privacy Act of 1986 (ECPA), 18 U.S.C. 2510-22. URL ⟨https://it.ojp.gov/default.aspx?area=privacy&page=1285⟩.

Doychev, G., Feld, D., Köpf, B., Mauborgne, L., Reineke, J., 2013. CacheAudit: a tool for the static analysis of cache side channels. In: Proceedings of the 22nd USENIX Conference on Security, SEC'13. USENIX Association, Berkeley, CA, USA, pp. 431–446.

ElGamal, T., 2006. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory 31 (4), 469–472.

Ertaul, L., Singhal, S., Saldamli, G., 2010. Security challenges in cloud computing. In: Proceedings of the 2010 International Conference on Security& Management, SAM 2010, July 12–15, 2010, Las Vegas Nevada, USA, 2 vols., pp. 36–42.

Fernandez, E.B., Monge, R., Hashizume, K., 2013. Two patterns for cloud computing: secure virtual machine image repository and cloud policy management point. In: Proceedings of the 20th Conference on Pattern Languages of Programs, PLoP '13. The Hillside Group, USA, pp. 15:1–15:11.

FireHost, 2015. True ePHI Protection in the Secure Cloud. URL ⟨https://www.firehost.com/company/contact⟩.

Fong, E., Okun, V., 2007. Web application scanners: definitions and functions. In: Proceedings of the 40th Annual Hawaii International Conference on System Sciences, HICSS '07. IEEE Computer Society, Washington, DC, USA, p. 280b.

Ken, E., Ramarathnam, V., 2012. FPGAs for Trusted Cloud Computing, IEEE. URL ⟨http://research.microsoft.com/apps/pubs/default.aspx?id=170502⟩.

FTC, 2015. Fair Credit Reporting Act. URL ⟨https://www.ftc.gov/enforcement/rules/rulemaking-regulatory-reform-proceedings/fair-credit-reporting-act⟩.

Fu, J., Wang, C., Yu, Z., Wang, J., Sun, J.-G., 2010. A watermark-aware trusted running environment for software clouds. In: 2010 Fifth Annual ChinaGrid Conference (ChinaGrid), pp. 144–151. http://dx.doi.org/10.1109/ChinaGrid.2010.15.

Futral, W., Greene, J., 2013. Intel Trusted Execution Technology for Server Platforms: A Guide to More Secure Datacenters, 1st edition. Apress, Berkely, CA, USA.

Garfinkel, T., Rosenblum, M., 2005. 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 – Volume 10, HOTOS'05. USENIX Association, Berkeley, CA, USA, pp. 20–20. URL ⟨http://dl.acm.org/citation.cfm?id=1251123.1251143⟩.

Godfrey, M., Zulkernine, M., 2013. A server-side solution to cache-based side-channel attacks in the cloud. In: 2013 IEEE Sixth International Conference on Cloud Computing (CLOUD), pp. 163–170. http://dx.doi.org/10.1109/CLOUD.2013.21.

GPO, 2015. Gramm-Leach-Bliley Act. URL ⟨http://legislink.org/us/stat-113-1338⟩.

Grobauer, B., Walloschek, T., Stocker, E., 2011. Understanding cloud computing vulnerabilities. IEEE Secur. Privacy 9 (2), 50–57.

Gruschka, N., Iacono, L., 2009. Vulnerable cloud: soap message security validation revisited. In: IEEE International Conference on Web Services, 2009. ICWS 2009, pp. 625–631. http://dx.doi.org/10.1109/ICWS.2009.70.

Gruschka, N., Jensen, M., 2010. Attack surfaces: a taxonomy for attacks on cloud services. In: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 276–279. http://dx.doi.org/10.1109/CLOUD.2010.23.

Harnik, D., Pinkas, B., Shulman-Peleg, A., 2010. Side channels in cloud services: deduplication in cloud storage. IEEE Secur. Privacy 8 (6), 40–47. http://dx.doi.org/10.1109/MSP.2010.187.

Hashizume, K., Yoshioka, N., Fernandez, E.B., 2011. Misuse patterns for cloud computing. In: Proceedings of the 2nd Asian Conference on Pattern Languages of Programs, AsianPLoP '11. ACM, New York, NY, USA, pp. 12:1–12:6. http://dx.doi.org/10.1145/2524629.2524644. URL ⟨http://doi.acm.org/10.1145/2524629.2524644⟩.

Hashizume, K., Rosado, D., Fernndez-Medina, E., Fernandez, E., 2013. An analysis of security issues for cloud computing. J. Internet Serv. Appl. 4 (1). http://dx.doi.org/10.1186/1869-0238-4-5.

Hay, A., Cid, D., Bary, R., Northcutt, S., 2008. In: {OSSEC} Host-Based Intrusion Detection Guide. Syngress, Burlington, pp. 247–249.

Hlavacs, H., Treutner, T., Gelas, J.-P., Lefevre, L., Orgerie, A.-C., 2011. Energy consumption side-channel attack at virtual machines in a cloud. In: 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), pp. 605–612. http://dx.doi.org/10.1109/DASC.2011.110.

Hwang, K., Li, D., 2010. Trusted cloud computing with secure resources and data coloring. IEEE Internet Comput. 14 (5), 14–22. http://dx.doi.org/10.1109/MIC.2010.86.

IBM, 2014. The 2013 IBM cyber security intelligence index. URL ⟨http://www-935.ibm.com/services/us/en/security/infographic/cybersecurityindex.html⟩, April.

IBM-Inc., 2015. IBM System z: Linux on System z-Solutions – Security Certification. URL ⟨http://www-03.ibm.com/systems/z/os/linux/solutions/security_certification.html⟩.

InfoSecurity, 2009. Google Cloud Platform Used for Botnet Control. URL ⟨http://www.infosecurity-magazine.com/news/google-cloud-platform-used-for-botnet-control/⟩.

Intel, 2013. Building Trust and Compliance in the Cloud with Intel Trusted Execution Technology. URL ⟨http://www.hytrust.com/sites/default/files/cloud-computing-txt-xeon-twse-whitepaper.pdf⟩, October.

Jansen, W.A., 2011. Cloud hooks: security and privacy issues in cloud computing. In: Proceedings of the 2011 44th Hawaii International Conference on System Sciences, HICSS '11. IEEE Computer Society, Washington, DC, USA, pp. 1–10.

Jansen, B., Ramasamy, H.-G.V., Schunter, M., 2008. Policy enforcement and compliance proofs for xen virtual machines. In: Proceedings of the Fourth ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, VEE '08. ACM, New York, NY, USA, pp. 101–110. http://dx.doi.org/10.1145/1346256.1346271. URL ⟨http://doi.acm.org/10.1145/1346256.1346271⟩.

Jaquier, C., 2013. Fail2ban Manual 0.8. URL ⟨http://www.fail2ban.org/wiki/index.php/MANUAL_0_8⟩.

Jasti, A., Shah, P., Nagaraj, R., Pendse, R., 2010. Security in multi-tenancy cloud. In: 2010 IEEE International Carnahan Conference on Security Technology (ICCST), pp. 35–41. http://dx.doi.org/10.1109/CCST.2010.5678682.

Jung, T., Li, X.-Y., Wan, Z., Wan, M., 2013. Privacy preserving cloud data access with multi-authorities. In: 2013 Proceedings IEEE INFOCOM, pp. 2625–2633. http://dx.doi.org/10.1109/INFCOM.2013.6567070.

Kaaniche, N., Laurent, M., 2014. A secure client side deduplication scheme in cloud storage environments. In: 2014 6th International Conference on New Technologies, Mobility and Security (NTMS), pp. 1–7. http://dx.doi.org/10.1109/NTMS.2014.6814002.

Karnwal, T., Sivakumar, T., Aghila, G., 2012. A comber approach to protect cloud computing against xml ddos and http ddos attack. In: 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS), pp. 1–5. http://dx.doi.org/10.1109/SCEECS.2012.6184829.

Khalil, I.M., Khreishah, A., Azeem, M., 2014. Cloud computing security: a survey. Computers 3, 1–35.

Khorshed, M.T., Ali, A.S., Wasimi, S.A., 2012. 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.

Ko, R., Lee, S.G., Rajan, V., 2013. Cloud Computing Vulnerability Incidents: A Statistical Overview. URL ⟨https://cloudsecurityalliance.org/download/cloud-computing-vulnerability-incidents-a-statistical-overview/⟩, March.

Kourai, K., Azumi, T., Chiba, S., 2012. A self-protection mechanism against stepping-stone attacks for iaas clouds. In: 2012 9th International Conference on Ubiquitous Intelligence Computing and 9th International Conference on Autonomic Trusted Computing (UIC/ATC), pp. 539–546. http://dx.doi.org/10.1109/UIC-ATC.2012.139.

Kumar, K., Liu, J., Lu, Y.-H., Bhargava, B., 2013. A survey of computation offloading for mobile systems. Mob. Netw. Appl. 18 (1), 129–140.

Leandro, M.A.P., Nascimento, T.J., dos Santos, D.R., Westphall, C.M., Westphall, C.B., 2012. Multi-tenancy authorization system with federated identity for cloud-based environments using Shibboleth. In: Proceedings of the Eleventh International Conference on Networks, pp. 88–95.

Lewis, G., 2013. Role of standards in cloud-computing interoperability. In: 2013 46th Hawaii International Conference on System Sciences (HICSS), pp. 1652–1661. http://dx.doi.org/10.1109/HICSS.2013.470.

Li, X.-Y., Zhou, L.-T., Shi, Y., Guo, Y., 2010. A trusted computing environment model in cloud architecture. In: 2010 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 6, pp. 2843–2848. http://dx.doi.org/10.1109/ICMLC.2010.5580769.

Li, M., Yu, S., Ren, K., Lou, W., Hou, Y., 2013. Toward privacy-assured and searchable cloud data storage services. IEEE Netw. 27 (4), 56–62. http://dx.doi.org/10.1109/MNET.2013.6574666.

Lin, W., Lee, D., 2012. Traceback attacks in cloud – pebbletrace botnet. In: 2012 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 417–426. http://dx.doi.org/10.1109/ICDCSW.2012.61.

Lindqvist, U., Porras, P.A., 2001. eXpert-BSM: a host-based intrusion detection solution for sun solaris. In: Proceedings of 17th Annual Computer Security Applications Conference. IEEE Computer Society, New Orleans, Louisiana, pp. 240–251.

Liu, S.-T., Chen, Y.-M., 2010. Retrospective detection of malware attacks by cloud computing. In: 2010 International Conference on Cyber-Enabled Distributed

Computing and Knowledge Discovery (CyberC), pp. 510–517. http://dx.doi.org/10.1109/CyberC.2010.99.

Liu, B., Xu, E., Wang, J., Wei, Z., Xu, L., Zhao, B., Su, J., 2011. Thwarting audio steganography attacks in cloud storage systems. In: 2011 International Conference on Cloud and Service Computing (CSC), pp. 259–265. http://dx.doi.org/10.1109/CSC.2011.6138530.

Liu, F., Ren, L., Bai, H., 2014. Mitigating cross-vm side channel attack on multiple tenants cloud platform. J. Comput. 9 (4).

Maler, E., Reed, D., 2008. The venn of identity: options and issues in federated identity management. IEEE Secur. Privacy 6 (2), 16–23.

Manuel, P., Thamarai Selvi, S., Barr, M.-E., 2009. Trust management system for grid and cloud resources. In: First International Conference on Advanced Computing, 2009. ICAC 2009, pp. 176–181. http://dx.doi.org/10.1109/ICADVC.2009.5378187.

Martignoni, L., Paleari, R., Bruschi, D., 2009. A framework for behavior-based malware analysis in the cloud. In: Proceedings of the 5th International Conference on Information Systems Security, ICISS '09. Springer-Verlag, Berlin, Heidelberg, pp. 178–192.

Mazurczyk, W., Szczypiorski, K., 2011. Is cloud computing steganography-proof?. In: 2011 Third International Conference on Multimedia Information Networking and Security (MINES), pp. 441–442. http://dx.doi.org/10.1109/MINES.2011.95.

McMillan, R., 2009. Hackers Find a Home in Amazon EC2's Cloud. URL ⟨http://www.computerworld.com/article/2521744/security0/hackers-find-a-home-in-amazon-s-ec2-cloud.html⟩.

Mearian, L., 2013. No, Your Data isn't Secure in the Cloud. URL ⟨http://www.computerworld.com/article/2483552/cloud-security/no-your-data-isn-t-secure-in-the-cloud.html⟩, August.

Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., Rajarajan, M., 2013. A survey of intrusion detection techniques in cloud. J. Netw. Comput. Appl. 36 (1), 42–57.

Morsy, M.A., Grundy, J., Muller, I., 2010. An analysis of the cloud computing security problem. In: Proceedings of APSEC 2010 Cloud Workshop, Sydney, Australia, pp. 1–6.

Murray, D.G., Milos, G., Hand, S., 2008. Improving xen security through disaggregation. In: Proceedings of the Fourth ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, VEE '08. ACM, New York, NY, USA, pp. 151–160. http://dx.doi.org/10.1145/1346256.1346278. URL ⟨http://doi.acm.org/10.1145/1346256.1346278⟩.

Oberheide, J., Cooke, E., Jahanian, F., 2008a. Cloudav: N-version antivirus in the network cloud. In: Proceedings of the 17th Conference on Security Symposium, SS'08. USENIX Association, Berkeley, CA, USA, pp. 91–106.

Oberheide, J., Cooke, E., Jahanian, F., 2008b. Empirical Exploitation Of Live Virtual Machine Migration, February.

OISF, 2015. Suricata User Guide. URL ⟨https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Suricata_User_Guide⟩.

OSSEC, 2015. OSSEC: How it Works. URL ⟨http://www.ossec.net/?page_id=169⟩.

Osvik, D.A., Shamir, A., Tromer, E., 2006. Cache attacks and countermeasures: the case of aes. In: Proceedings of the 2006 Cryptographers' Track at the RSA Conference on Topics in Cryptology, CT-RSA'06. Springer-Verlag, Berlin, Heidelberg, pp. 1–20.

OWASP, 2015a. Web Application Firewall. URL ⟨https://www.owasp.org/index.php/Web_Application_Firewall⟩.

OWASP, 2015b. Vulnerability Scanning Tools – OWASP. URL ⟨https://www.owasp.org/index.php/Category:Vulnerability⟩.

Owens, K., 2009. Securing virtual compute infrastructure in the cloud. In: White Paper: Cloud Computing. Savvis, Missouri, United States, pp. 1–13.

Owens, D., 2010. Securing elasticity in the cloud. Queue 8 (5) 10:10–10:16.

Paxson, V., 1999. Bro: a system for detecting network intruders in real-time. Comput. Netw. 31 (23-24), 2435–2463.

Popovic, O., Jovanovic, Z., Jovanovic, N., Popovic, R., 2011. A comparison and security analysis of the cloud computing software platforms. In: 2011 10th International Conference on Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS), vol. 2, pp. 632–634.

Ragouzis, N., Hughes, J., Philpott, R., Maler, E., Madsen, P., Scavo, T., 2008. Security Assertion Markup Language (SAML) V2.0 Technical Overview. URL ⟨http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf⟩, March.

Ranjith, P., Priya, C., Shalini, K., 2012. On covert channels between virtual machines. J. Comput. Virol. 8 (3), 85–97. http://dx.doi.org/10.1007/s11416-012-0168-x.

RedHat-Inc., 2015. Red Hat and IBM Achieve Top Security Certification for KVM Hypervisor on Red Hat Enterprise Linux and IBM Servers. URL ⟨http://www.redhat.com/en/about/press-releases/Red-Hat-and-IBM-Achieve-Top-Security-Certification-for-KVM-Hypervisor-on-Red-Hat-Enterprise-Linux-and-IBM-Servers⟩.

Reimer, D., Thomas, A., Ammons, G., Mummert, T., Alpern, B., Bala, V., 2008. Opening black boxes: using semantic information to combat virtual machine image sprawl. In: Proceedings of the Fourth ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, VEE '08. ACM, New York, NY, USA, pp. 111–120.

Riquet, D., Grimaud, G., Hauspie, M., 2012. Large-scale coordinated attacks: impact on the cloud security. In: 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 558–563. http://dx.doi.org/10.1109/IMIS.2012.76.

Ristenpart, T., Tromer, E., Shacham, H., Savage, S., 2009. 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, CCS '09. ACM, New York, NY, USA, pp. 199–212. http://dx.doi.org/10.1145/1653662.1653687. URL ⟨http://doi.acm.org/10.1145/1653662.1653687⟩.

Rocha, F., Correia, M., 2011. Lucy in the sky without diamonds: stealing confidential data in the cloud. In: Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops, DSNW '11. IEEE Computer Society, Washington, DC, USA, pp. 129–134. http://dx.doi.org/10.1109/DSNW.2011.5958798. URL ⟨http://dx.doi.org/10.1109/DSNW.2011.5958798⟩.

Roesch, M., 2014. Snort User Manual 2.9.7. URL ⟨https://www.snort.org/documents/1⟩, October.

Rong, H., Xian, M., Wang, H., Shi, J., 2013. Time-stealer: a stealthy threat for virtualization scheduler and its countermeasures. In: Qing, S., Zhou, J., Liu, D. (Eds.), Information and Communications Security, Lecture Notes in Computer Science vol. 8233. Springer International Publishing, Tainan, Taiwan, pp. 100–112.

Rueda, S., Sreenivasan, Y., Jaeger, T., 2008. Flexible security configuration for virtual machines. In: Proceedings of the 2nd ACM Workshop on Computer Security Architectures, CSAW '08. ACM, New York, NY, USA, pp. 35–44. http://dx.doi.org/10.1145/1456508.1456515. URL ⟨http://doi.acm.org/10.1145/1456508.1456515⟩.

Sagan, 2015. The Sagan Log Analysis Engine. URL ⟨http://sagan.quadrantsec.com/⟩.

Samhain, 2006. The Samhain File Integrity/Host-Based Intrusion Detection System. URL ⟨http://www.la-samhna.de/samhain/⟩.

Sanaei, Z., Abolfazli, S., Gani, A., Buyya, R., 2014. Heterogeneity in mobile cloud computing: taxonomy and open challenges. IEEE Commun. Surv. Tutor. 16 (1), 369–392. http://dx.doi.org/10.1109/SURV.2013.050113.00090.

Sanchez, R., Almenares, F., Arias, P., Diaz-Sanchez, D., Marin, A., 2012. Enhancing privacy and dynamic federation in IdM for consumer cloud computing. IEEE Trans. Consum. Electron. 58 (1), 95–103.

Santos, N., Gummadi, K.P., Rodrigues, R., 2009. Towards trusted cloud computing. In: Proceedings of the 2009 Conference on Hot Topics in Cloud Computing, HotCloud'09. USENIX Association, Berkeley, CA, USA. URL ⟨http://dl.acm.org/citation.cfm?id=1855533.1855536⟩.

Scarfone, K.A., Mell, P.M., 2007. Sp 800-94. Guide to Intrusion Detection and Prevention Systems (IDPS). Technical Report, Gaithersburg, MD, United States.

Sen, J., 2013. Security and privacy issues in cloud computing, CoRR abs/1303.4814. URL ⟨http://arxiv.org/abs/1303.4814⟩.

Shahzad, F., 2014. State-of-the-art survey on cloud computing security challenges, approaches and solutions. Proc. Comput. Sci. 37, 357–362, The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014)/The 4th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2014)/Affiliated Workshops.

Shankarwar, M., Pawar, A., 2015. Security and privacy in cloud computing: a survey. In: Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014, Advances in Intelligent Systems and Computing, vol. 328. Springer International Publishing, Bhubaneswar, Odisha, India, pp. 1–11.

Shen, Z., Li, L., Yan, F., Wu, X., 2010. Cloud computing system based on trusted computing platform. In: 2010 International Conference on Intelligent Computation Technology and Automation (ICICTA), vol. 1, pp. 942–945. http://dx.doi.org/10.1109/ICICTA.2010.724.

Somani, U., Lakhani, K., Mundra, M., 2010a. Implementing digital signature with rsa encryption algorithm to enhance the data security of cloud in cloud computing. In: 2010 1st International Conference on Parallel Distributed and Grid Computing (PDGC), pp. 211–216. http://dx.doi.org/10.1109/PDGC.2010.5679895.

Somani, U., Lakhani, K., Mundra, M., 2010b. Implementing digital signature with rsa encryption algorithm to enhance the data security of cloud in cloud computing. In: 2010 1st International Conference on Parallel Distributed and Grid Computing (PDGC), pp. 211–216. http://dx.doi.org/10.1109/PDGC.2010.5679895.

Srinivasamurthy, S., Liu, D.Q., Vasilakos, A.V., Xiong, N., 2013. Cloud computing security: a survey. Parallel Cloud Comput. 2 (4), 126–153.

Stefanov, E., Shi, E., 2013. Oblivistore: high performance oblivious cloud storage. In: 2013 IEEE Symposium on Security and Privacy (SP), pp. 253–267. http://dx.doi.org/10.1109/SP.2013.25.

Stolfo, S., Salem, M., Keromytis, A., 2012. Fog computing: mitigating insider data theft attacks in the cloud. In: 2012 IEEE Symposium on Security and Privacy Workshops (SPW), pp. 125–128.

Su, T.-A., 2013. A mechanism to prevent side channel attacks in cloud computing environments. In: 2013 World Congress in Computer Science, Computer Engineering and Applied Computing.

Subashini, S., Kavitha, V., 2011. A survey on security issues in service delivery models of cloud computing. J. Netw. Comput. Appl. 34 (1), 1–11.

Symantec, 2015. 2015 Internet Security Threat Report. URL ⟨http://www.symantec.com/security_response/publications/threatreport.jsp⟩, April.

Symform, 2014. Achieving HIPAA Compliant Cloud Backup. URL ⟨http://www.symform.com/hipaa-compliance/⟩.

Szefer, J., Lee, R.B., 2012. Architectural support for hypervisor-secure virtualization. SIGARCH Comput. Archit. News 40 (1), 437–450.

Takabi, H., Joshi, J., Ahn, G.-J., 2010. Securecloud: towards a comprehensive security framework for cloud computing environments. In: 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops (COMPSACW), pp. 393–398. http://dx.doi.org/10.1109/COMPSACW.2010.74.

Tandon, S., SB, S., Agrawal, V., 2014. Cache-based side-channel attack on aes in cloud computing environment. Int. J. Eng. Res. Technol. 3 (10), 1080–1084.

TCG, 2015. Trusted Computing Group. URL ⟨http://www.trustedcomputinggroup.org/⟩.

TCG, 2015. Cloud Computing and Security: A Natural Match. URL ⟨http://www.trustedcomputinggroup.org/⟩.

Tcpdump, 2015. Tcpdump & libpcap. URL ⟨http://www.tcpdump.org/⟩.

Tebaa, M., El Hajji, S., El Ghazi, A., 2012. Homomorphic encryption method applied to cloud computing. In: 2012 National Days of Network Security and Systems (JNS2), pp. 86–89. http://dx.doi.org/10.1109/JNS2.2012.6249248.

Townsend, M., 2009. Managing a security program in a cloud computing environment. In: 2009 Information Security Curriculum Development Conference, InfoSecCD '09. ACM, New York, NY, USA, pp. 128–133. http://dx.doi.org/10.1145/1940976.1941001. URL ⟨http://doi.acm.org/10.1145/1940976.1941001⟩.

Trusted Computing Group, 2011. TPM Main Specification. URL ⟨http://www.trustedcomputinggroup.org/resources/tpm_main_specification⟩.

U.D. of Health & Human Services, 2007. Health Information Policy. URL ⟨http://www.hhs.gov/ocr/privacy/hipaa/administrative/privacyrule/index.html⟩.

Vasudevan, A., McCune, J.M., Qu, N., Van Doorn, L., Perrig, A., 2010. Requirements for an integrity-protected hypervisor on the x86 hardware virtualized architecture. In: Proceedings of the 3rd International Conference on Trust and Trustworthy Computing, TRUST'10. Springer-Verlag, Berlin, Heidelberg, pp. 141–165. URL ⟨http://dl.acm.org/citation.cfm?id=1875652.1875663⟩.

VMWare, 2015. VMWare Certifications – Common Criteria Security Certification. URL ⟨https://www.vmware.com/security/certifications/common-criteria⟩.

Wang, Z., Jiang, X., 2010. Hypersafe: a lightweight approach to provide lifetime hypervisor control-flow integrity. In: 2010 IEEE Symposium on Security and Privacy (SP), pp. 380–395. http://dx.doi.org/10.1109/SP.2010.30.

Wang, Z., Lee, R.B., 2007. New cache designs for thwarting software cache-based side channel attacks. SIGARCH Comput. Archit. News 35 (2), 494–505.

Wang, C., Wang, Q., Ren, K., Lou, W., 2009. Ensuring data storage security in cloud computing. In: 17th International Workshop on Quality of Service, 2009. IWQoS, pp. 1–9. http://dx.doi.org/10.1109/IWQoS.2009.5201385.

WCSS, 2010. Acarmng User Manual. URL ⟨http://www.acarm.wcss.wroc.pl/index.php?n=Acarmng.Doc⟩.

Wei, J., Zhang, X., Ammons, G., Bala, V., Ning, P., 2009. Managing security of virtual machine images in a cloud environment. In: Proceedings of the 2009 ACM Workshop on Cloud Computing Security, CCSW '09. ACM, New York, NY, USA, pp. 91–96.

Winkler, J.R.V., 2011. Securing the Cloud: Cloud Computer Security Techniques and Tactics. Syngress Publishing, Massachusetts, USA.

Wu, H., Ding, Y., Winer, C., Yao, L., 2010. Network security for virtual machine in cloud computing. In: 2010 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT), pp. 18–21. http://dx.doi.org/10.1109/ICCIT.2010.5711022.

Wu, J., Ding, L., Wu, Y., Min-Allah, N., Khan, S.U., Wang, Y., 2014. C^2detector: a covert channel detection framework in cloud computing. Secur. Commun. Netw. 7 (3), 544–557.

Wylie, J.J., Bakkaloglu, M., Pandurangan, V., Bigrigg, M.W., Oguz, S., Tew, K., Williams, C., Ganger, G.R., Khosla, P.K., 2001. Selecting the right data distribution scheme for a survivable storage system. In: CMU-CS-01-120, pp. 1–23.

Xiao, S., Gong, W., 2010. Mobility can help: protect user identity with dynamic credential. In: 2010 Eleventh International Conference on Mobile Data Management (MDM), pp. 378–380.

Xiaopeng, G., Sumei, W., Xianqin, C., 2010. VNSS: a network security sandbox for virtual computing environment. In: 2010 IEEE Youth Conference on Information Computing and Telecommunications (YC-ICT), pp. 395–398. http://dx.doi.org/10.1109/YCICT.2010.5713128.

Yang, Z., Qiao, L., Liu, C., Yang, C., Wan, G., 2010. A collaborative trust model of firewall-through based on cloud computing. In: 2010 14th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 329–334. http://dx.doi.org/10.1109/CSCWD.2010.5471954.

Zhang, F., Huang, Y., Wang, H., Chen, H., Zang, B., 2008. Palm: security preserving vm live migration for systems with VMM-enforced protection. In: Third Asia-Pacific Trusted Infrastructure Technologies Conference, 2008. APTC '08, pp. 9–18. http://dx.doi.org/10.1109/APTC.2008.15.

Zhang, Y., Juels, A., Reiter, M.K., Ristenpart, T., 2012. Cross-vm side channels and their use to extract private keys. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS '12. ACM, New York, NY, USA, pp. 305–316. http://dx.doi.org/10.1145/2382196.2382230. URL ⟨http://doi.acm.org/10.1145/2382196.2382230⟩.

Zhang, Y., Juels, A., Reiter, M.K., Ristenpart, T., 2014. Cross-tenant side-channel attacks in paas clouds. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, CCS '14. ACM, New York, NY, USA, pp. 990–1003. http://dx.doi.org/10.1145/2660267.2660356. URL ⟨http://doi.acm.org/10.1145/2660267.2660356⟩.

Zhou, M., Zhang, R., Xie, W., Qian, W., 2010. A. Zhou, Security and privacy in cloud computing: a survey. In: 2010 Sixth International Conference on Semantics Knowledge and Grid (SKG), pp. 105–112.

Zhou, F., Goel, M., Desnoyers, P., Sundaram, R., 2011. Scheduler vulnerabilities and coordinated attacks in cloud computing. In: 2011 10th IEEE International Symposium on Network Computing and Applications (NCA), pp. 123–130. http://dx.doi.org/10.1109/NCA.2011.24.

Zissis, D., Lekkas, D., 2012. Addressing cloud computing security issues. Future Gener. Comput. Syst. 28 (3), 583–592.