JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 611-634 (2017)
DOI: 10.6688/JISE.2017.33.3.2

Intrusion Detection Systems in Cloud Computing: A Contemporary Review of Techniques and Solutions*

AMNA RIAZ[1], HAFIZ FAROOQ AHMAD[2], ADNAN KHAILD KIANI[1], JUNAID QADIR[3], RAIHAN UR RASOOL[1,2,5,+] AND USMAN YOUNIS[1,4]
[1]*School of Electrical Engineering and Computer Science*
National University of Sciences and Technology
Islamabad, 44000 Pakistan
[2]*Department of Computer Science*
King Faisal University
Al-Hofuf, 31982 Saudi Arabia
[3]*Information Technology University*
Lahore, 54000 Pakistan
[4]*Department of Electrical and Computer Engineering*
National University of Singapore
Singapore, 117583 Singapore
[5]*Victoria University*
Melbourne, 14428 Australia
E-mail: [+]rrasool@kfu.edu.sa, raihan.rasool@seecs.edu.pk

Rapid growth of resources and escalating cost of infrastructure is leading organizations to adopt cloud computing. Cloud computing provides high performance, efficient utilization, and on-demand availability of resources. However, the cloud environment is vulnerable to different kinds of intrusion attacks which involve installing malicious software and creating backdoors. In a cloud environment, where businesses have hosted important and critical data, the security of underlying technologies becomes crucial. To mitigate the threat to cloud environments, Intrusion Detection Systems (IDS) are a layer of defense. The aim of this survey paper is to review IDS techniques proposed for the cloud. To achieve this objective, the first step is defining the limitations and unique characteristics of each technique. The second step is establishing the criteria to evaluate IDS architectures. In this paper, the criteria used is derived from basic characteristics of cloud. Next step is a comparative analysis of various existing intrusion detection techniques against the criteria. The last step is on the discussion of drawbacks and open issues, comprehended from the evaluation, due to which implementation of IDS in cloud environment face hurdles.

Keywords: intrusion detection systems, cyber-security, cloud computing, comparative analysis, open issues

1. INTRODUCTION

Cloud computing is an emerging technology adopted by organizations of all scale due to its low-cost and pay-as-you-go structure. It has revolutionized the IT world with its unique and ubiquitous capabilities. Organization prefers cloud as it replaces the high price infrastructure and need of maintenance. It offers three service models of software as a service (*e.g.* Google Apps [1]), platform as a service (*e.g.* Google App Engine [2],

Received June 26, 2016; accepted October 22, 2016.
Communicated by Ce-Kuen Shieh.
* Authors extend their sincere appreciation to the Deanship of Scientific Research at King Faisal University for funding this research work through annual grant number 160088.

Microsoft's Azure [3]) and infrastructure as a service (*e.g.* Amazon Web Service [4], Eucalyptus [5], Open Nebula [6]). Virtualization enables cloud to provide elasticity, ease of use, scalability and on-demand network access to a shared pool of configurable computing resources [7]. Cloud computing paradigm has a service-oriented architecture which has led to a drastic alteration on how services are provided and managed.

Intrusion detection techniques are used in any computing environment as a layer of defense. The basic aim is to detect any malicious activity well before any significant harm is possible. The general idea is to detect and identify attacks by either analyzing system artifacts (such as log files, process lists, *etc.*), or by keeping track of network traffic. Two main approaches used are signature based detection and anomaly-based detection. Signature based detection works by defining patterns of known attack signatures. If the system is found to be processing any code similar to those signatures, it is detected suspicious and marked as an intrusion. On the other hand, anomaly based detection works by analyzing activities performed on the system. Initially, a profile for a particular system is created by recording normal activities (*e.g.*, by setting thresholds for normal bandwidth usage). If later on, the system's behavior is analyzed as anomalous to the profile defined, it is marked as an intrusion. Whereas signature-based detection techniques (also called misuse pattern matching) cannot detect unknown attacks, anomaly based techniques usually result in huge false positives or negatives.

The distributed nature of cloud environment makes it most vulnerable and attractive environment for the intruders to perform attacks. Intrusion detection systems can be used to enhance the security of such systems by systematically examining the logs, network traffic as well as configurations. However conventional intrusion detection systems (IDSs) – which can be classified into host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS) – are not appropriate for cloud environment as these are unable to locate the hidden attack trail, *e.g.*, the network-based IDS is unable to detect any event in case of encrypted node communication and it is possible for the attacker to gain control over the installed virtual machines if the hypervisor is compromised. Some of the popular attacks on virtual machine include DKSM [8], SubVirt [9], and Bluepill [10]. Attackers can use the compromised hypervisor to gain control over the host. Owing to the fact that the IDS techniques were not designed with the specific context of virtualization under consideration, they do not offer the same protection in such environments. There are certain trade-offs that need to be faced when deploying IDS in the virtual environment, mostly because of their inability to inspect the internal working of the operating systems. Despite the huge benefits that are offered by virtualization, there are a number of security risks that are associated with it. It introduces a number of new problems that did not exist in a traditional computing environment.

Cloud computing providers are adopting software-defined networking (SDN) to achieve on-demand provisioning of network services, since SDN can provide a centralized system to manage the network. The network administrator is empowered by SDN to easily access and manage individual flows by facilitating them to implement monitoring applications, *i.e.*, firewall and IDS. Furthermore, scalable monitoring and dynamic reconfiguration requirements of the network in cloud makes SDN a perfect choice.

This paper analyzes various IDS techniques proposed in literature on the basis of a set of requirements (essentially drawing from the list of requirements articulated by Patel *et al.* [11], and supplemented by an additional postulate that we propose). The concept of

deploying SDN in the cloud was not implemented at the time Patel *et al.* proposed the list. Although the implementation of SDN in cloud environment is in its initial phases, but the cloud vendors have started to adopt SDN to achieve their networking requirements. The additional requirement, which we have placed for analysis, is the effective working of IDS in a SDN-based cloud environment. Finally, we have discussed virtual machine introspection (VMI) based techniques in detail.

A few number of survey papers targeting this domain already exist. However, most of these survey papers are outdated or stinting in their coverage on cloud-based IDS. Zbakh *et al.* [12] proposed a multi-criteria analysis and a comparative study of several IDS architectures designed to work in cloud computing. It is the only research paper that uses a multi-criteria decision analysis – which the authors have named MacBeth (Measuring Attractiveness by a Categorical Based Evaluation Technique) – to evaluate IDS architectures in cloud. Modi *et al.* [13] and Mehmood *et al.* [14] have surveyed different intrusions in cloud computing which affects the confidentiality, integrity, and availability (CIA triad) of cloud environment. An analysis of IDS and intrusion prevention system (IPS) techniques is performed. Furthermore, Modi *et al.* emphasize that in order to reach the desired security level, deployment position of IDS is crucial. Oktay *et al.* [15] enlist the attack types in cloud followed by details of IDS models to resist them. Patel *et al.* [11] and Premathilaka *et al.* [16] stress on designing IDS especially for cloud environment keeping in mind its paradigm after conducting a review and highlighting how traditional IDS fail to deliver. Moreover, Patel *et al.* provide us a list of requirements derived from the characteristics of cloud computing systems proposed by NIST [7] in order to analyze cloud-based intrusion detection or prevention system; techniques till mid-2012 have been discussed and comparatively analyzed. A tabulated summary, which presents the comparison of this survey paper with the existing surveys in literature, is given in Table 1.

Table 1. Coverage of various topics in the existing surveys.

Survey	Performance evaluation	Advantages & disadvantages of each approach	Intrusions in clouds	Open issues
[13]	×	✓	✓	×
[14]	×	✓	✓	×
[16]	×	×	×	×
[11]	✓	✓	×	×
[15]	×	×	✓	×
[12]	✓	×	×	×
This Research Paper	✓ (comparatively analyzed)	✓	✓	✓

In this paper, we have discussed in particular the gaps which require further study and have provided a comprehensive and contemporary review of various IDS techniques. Furthermore, prospective areas of studies have been discussed which require attention of the researchers in order to improve the implementation of current IDSs in cloud environment.

The paper is organized as follow. Section 2 highlights the various intrusion/attacks in cloud. In section 3, summary of the existing techniques for IDSs in the cloud is pre-

sented. The comparative performance analysis of all these IDS techniques is presented in section 4. In section 5, the common hurdles faced during the deployment of a cloud IDS are pointed out, and various open research issues are highlighted. Finally, we conclude our work and give directions for further study in section 6.

2. INTRUSIONS IN CLOUD

An attempt to compromise the confidentiality, integrity, or availability of a system or network is known as an intrusion. In this section important classes of intrusion that commonly affect the cloud are described. This is followed by a presentation of various attacks in the cloud, classified with respect to cloud's deployment model.

2.1 Denial of Service (DoS) Attack

The hacker uses bots (zombies) for flooding a system with a large number of packets to render the available resources unreachable. Subsequently, the services for the time being are not available on the Internet. According to some vulnerability experts, an attacker can affect more users by launching a DoS attack on cloud [17].

2.2 Insider Attack

Insider is defined as a former or current employee/associate of the cloud service provider which has privileged access and authority to perform modifications in the cloud environment [18]. Insider attacks are organized as they have information about the user and provider. This is fatal as many attacks can be executed from inside and an intruder can easily evade detection in the absence of proper controllers [19, 20]. A DoS attack by an insider was launched on Amazon Elastic Compute Cloud (EC2) [21], cloud consumers' confidentiality was breached in this attack.

2.3 User to Root (U2R) Attack

In this attack, the intruder accesses the credentials of an authentic user and then exploit the system vulnerabilities (buffer overflow) to access root privileges. In the cloud, the attacker first accesses an instance and exploits its vulnerabilities to achieve root privileges of a virtual machine or host. By this attack, integrity of the cloud is being violated [13].

2.4 Port Scanning

Port scanning is used by the attacker to obtain information about open, closed, filtered, and unfiltered ports [13]. The attacker then uses this information to launch attacks on open ports. Different techniques are used in order to perform port scanning. This attack targets the confidentiality and integrity of the cloud.

2.5 Attacks on Virtualization

If an attacker compromises the hypervisor, the virtual machines can be easily infil-

trated [13]. The best option to capture virtual machines via hypervisor is to exploit a zero-day vulnerability. Zero-day attacks are exploitation of vulnerabilities for which system administrator or developer has not applied the patch. Since many virtual machines use the same resources, *i.e.* hardware, side channel data is vulnerable due to this type of access among virtual machines [22].

2.6 Backdoor Channel Attack

This is a passive attack in which a node in cloud is compromised and in future the node is used as a bot to carry out attacks like DDoS attack. The system is compromised by shellcode, Trojan, and other similar exploitations. After the node is compromised the intruder has full access to the system and data available [13].

In Fig. 1, intrusions have been identified and classified on the basis of deployment model (SaaS, PaaS and IaaS).

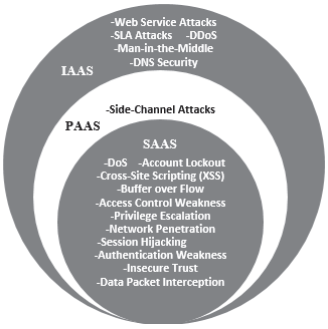


Fig. 1. Types of security attacks in cloud on the basis of deployment model.

3. STATE OF THE ART IDS ARCHITECTURES IN CLOUD

As mentioned in Section 1, conventional IDS *i.e.* HIDS and NIDS are not suited for virtual systems. The emerging threats to cloud security have led researchers to contribute a reasonable amount of work in the field of cloud-based IDS. In this paper, the architectures are divided and covered on the basis of deployment of IDS in cloud, *i.e.*, HIDS, NIDS, DIDS, and VMI Techniques, as shown in Fig. 2.

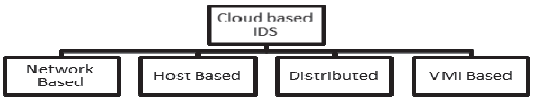


Fig. 2. Types of Cloud-Based IDS.

All of these deployment models can use anomaly detection or signature detection techniques. This section covers a survey of all these IDSs suited for the cloud.

3.1 Host Intrusion Detection Systems

HIDS techniques, with respect to cloud generally, can be divided into three main

deployment based classifications. HIDS, inside the VM for monitoring, can be deployed in the host OS (where it can monitor either the host OS or the guest OS via communication through the VMM [23]) or in a separate guest OS.

The first scenario, in which the IDS would be completely under the control of the customer, has the drawback of low attack resistance. It has been overwhelmingly rejected in the literature and hence marked unsuitable for the virtual cloud [24]. Laureano *et al.* [23] describe this as VMIs suited for type I or type II environment – where a type I environment implies that the VMM is the only process running on the host and several VMs run over it. A type II environment, on the other hand, implies that the VMM runs as the software on the host machine. Normal host processes as well as VMM (over which the VM) run on the host machine.

Patel *et al.* [25] propose an autonomic agent-based intrusion prevention using the principles of automatic computing. Anomaly based detection technique is used to monitor the system activities and network traffic via autonomous sensors for detection of malicious incidents. Lee *et al.* [26] propose a technique that detects suspicious behavior of an intrusion by anomaly level of resource utilization by the user. The main module of the technique is authentication, authorization, and accounting (AAA) component. The anomaly level is based on recent usage history of the user and stored in the database. IDS of medium level and low-level security utilizes fewer resources so more guest OS can be added without worrying about detection speed. The log files are available for the administrator for auditing.

Vieira *et al.* [27] propose an intrusion detection system for grid and cloud computing (GCCIDS). It is a combination of behavior based and knowledge based techniques at middleware layer to detect intrusions. It works in a cooperative manner where each node can detect intrusion and generate alerts for other nodes as well. Dhage *et al.* [28] propose that an IDS controller should install an IDS instance between cloud service provider (CSP) and user until the user is accessing cloud services. IDS controller collects the log files from all the IDS instances running. The log files help the controller maintain a knowledge-based for all the users based on their activities. It also helps IDS identify the user next time he/she logins. The drawback of knowledge based IDS is that it can only update new samples by the neural network so a workaround is proposed.

In Table 2 we have presented the characteristics and limitations of host-based IDS discussed above, such that these security challenges can be addressed before a standard technique is recommended for cloud. The ideal performance vs efficiency trade-off in host-IDS has not been achieved so far.

Table 2. Analysis of HIDS in cloud.

Research	Deployment	Technique	Characteristic	Limitation
[25]	Not specified	Anomaly based Detection	System activities (system calls *etc.*), Network traffic.	No implementation details
[26]	At each guest OS	Anomaly based Detection	User profiles, detect known attacks	High-level users consume more resources
[27]	At each node	Hybrid	Log files for auditing, user profiles	Accurate detection requires more time for training, number of rules is limited.
[28]	At each node	Hybrid	User activities logs, Can detect known attacks only	Experimental results are not given

Table 3. Analysis of NIDS in cloud.

Research	Deployment	Technique	Characteristic	Limitation
[34]	Inside MVM[1]	Signature based Detection	Can detect known attacks only	Multiple sensors, correlating their data can impact performance
[30]	vSwitch	Signature based Detection	SNORT for detecting DoS attacks	Can detect known attacks only, no support for large networks, misconfigurations in IDS will miss attacks [26]
[33]	PVM	Anomaly and Signature based Detection	Separate profile to increase efficiency	No prototype
[32]	At each node	Signature based Detection	Can detect known attacks only	Unable to detect unknown attacks

The technique presented in [23] is an example of the second scenario, where the HIDS runs as a host process but monitors the guest OS. However, this can also be classified as an HIDS-based VMI IDS and hence we discuss it later on in this section under the said category.

Similarly, the technique presented in [29] is an example of the third scenario. We will discuss this too under VMI-IDS.

3.2 Network Intrusion Detection Systems

Bakshi *et al.* [30] propose a typical NIDS for virtualized environments for the detection of DDoS attacks. A NIDS is installed on a virtual switch (through which traffic of all VMs collectively passes). This is analogous to a NIDS being placed at the boundary server in a traditional computing environment. The technique is very much similar to a traditional NIDS. However, the authors have tailored it suitable for the virtualized environment and tested it as such.

The NIDS on vSwitch collects inbound and outbound traffic from all VMs and logs it. It uses SNORT [31] tools to detect DoS or DDoS attack. Traffic is analyzed based on the originating IP address. Any IP address detected to send huge amount of abnormal traffic is blocked and the targeted application is moved to a different data center. The technique is capable of detecting DDoS attacks as well and blocking complete botnets. However, no performance results are discussed in the paper and the usage of SNORT for detection means that only known attacks are detectable [13]. Support for large virtual networks where traffic volume is quite huge is not discussed in this paper [16]. NIDS is going to face issues processing all packets in large virtual network and it may fail to detect attacks in time.

Mazzariello *et al.* [32] have simulated IDS at different locations in cloud to detect DoS attacks on virtual SIP-based hosts. It is a signature based detection technique. The Eucalyptus cloud computing environment has been used for experimentation and SNORT has been selected as the network IDS. Gupta *et al.* [33] address main limitations in previous techniques and propose a technique catering for those. The complexity introduced as a result of inspecting all VMs for all attacks. In their paper, they claim to

[1] MVM=Monitored VM; PVM= Privileged

mitigate this problem by introducing profile-based IDS. They propose an NIDS-based VMI for the cloud whereby a separate profile is created for each VM based on comparison with known attack signatures and deviation from normal thresholds.

The characteristics and limitations of network-based IDS have been enlisted in Table 3. The conclusion reached from the analysis is that the fast detection is a hurdle in network-based IDS. Furthermore, detection of unknown attacks is an issue since it further degrades the speed of the network.

3.3 Virtual Machine Introspection (VMI) Based Techniques

Virtual machine introspection (VMI) is the main idea behind out-of-box intrusion detection. VMI is a technique of inspecting VM state by moving the inspection module outside of the VM. The software running inside the guest system is analyzed externally to detect any intrusion. One advantage of this technique is that malware detection continues to work unaffectedly even in the presence of an intrusion. This capability is missing in HIDS and NIDS. In the case of a compromise, HIDS starts reporting falsely while NIDS has limited visibility [29]. Recently, the works of Garfinkel *et al.* [29], Jiang *et al.* [35], Laureano *et al.* [23], and Payne *et al.* [36] have demonstrated the use of VMI-based intrusion detection techniques. We discuss these techniques next.

Garfinkel *et al.* [29] propose Livewire, a prototype based on VMI for intrusion detection. This technique is based on the assumption that the VMM is simple and implemented correctly. This feature marks VMM safe and difficult for the attacker to compromise. The technique particularly leverages three properties of VMM, which are isolation, inspection, and interposition. The shortcoming in the research which demands further improvement is that the OS library interface in Livewire is required to be made in a safe programming language for different OSs to prevent an attack on itself. Livewire is considered to be revolutionary work in the domain of virtual machine IDS.

Jiang *et al.* [35] propose VMwatcher, which is an out-of-box deployment of intrusion detection with improved accuracy and tamper resistance. VMwatcher assumes the underlying VMM to be secure. The technique is applicable on type II VMs. The best claimed feature of VMwatcher is its ability to remove a semantic gap[2] that is bound to exist while gathering a view of guest OS externally. VMwatcher works on two techniques: 1) non-intrusive VM introspection, and 2) guest view casting. Non-intrusive introspection is same concept as VMI. Guest view casting is used to build the semantic view (*i.e.*, files, processes, directories, and kernel level modules) of VM. VM raw image, with memory states, is acquired, and the focus is to reconstruct a high level semantic view. The prototype is tested for both windows and Linux environment. VMwatcher maintains the VM view by carefully crafting the semantic view. Moreover, the authors claim that VMwatcher has the capability to detect stealthy malware by comparing the view generated internally with the one generated externally.

Another intrusion detection capability that is based on introspection is presented by Laureano *et al.* [23]. The technique is for type II VMs so that detection and response can be implemented as host system process. The interaction between a guest OS and an intrusion detection system takes place through a VMM. Two types of interactions are defined, including: 1) monitoring, in which the guest OS data is extracted, and 2) response,

[2] Semantic gap is the difference in views between the one generated from inside the VM and other generated from outside the VM.

in which the response as a result of malware detection is generated. The assumption is that VMM is inaccessible to the guest system's processes and is out of the reach of an attacker.

Payne *et al.* [36] propose the architecture of Lares, a prototype claimed to be more sound than XenAccess [37] (which was also proposed by the same authors). Lares is unique in terms of doing active monitoring – a feature missing in the previous VMI-based IDS. The performance analysis presents a tradeoff between the design (security) and the overhead in processing the placed hooks. Tupakula *et al.* [38] propose a VMM-based technique. The main issue discussed in the paper is of VM domains (*i.e.* VMs hosted on same hypervisor in one domain). It is stated that, in a cloud, different hypervisors host VMs of different consumers and different VMs of the same consumer can be hosted by several hypervisors within the cloud. They identify VMs allocated to the same consumer as one virtual domain, regardless of the hypervisor they are hosted on. Communication within a virtual domain could be completely out of control of the cloud service provider (CSP).

In the VMI-based IDS techniques, one important assumption is made about the security of underlying layer, *i.e.*, VMM/hypervisor is secure. The attacker might exploit a vulnerability in VMM and compromise guest OS kernel and finally the host kernel. In such a case there is a need to protect the VMM. Bharadwaja *et al.* [39] propose a technique that works by moving the IDS at the VMM layer to mitigate the threat. Collabra, a collaborative IDS, is a distributed platform based on Xen hypervisor to maintain the security of virtualized environment in cloud. Collabra is a filtering layer above hypervisor that performs an integrity check of the hyper call interface and works collaboratively with the distributed instances to detect and then prevent attacks. The objective is to protect the hyper call interface from attacks. The detection is performed based on anomaly detection. Collabra isolates the affected VMM from the network when an intrusion is detected.

Jones *et al.* [40] propose Lycosid, a VMM based intrusion detection that is based on cross view validation principle. Stealthy rootkits have the ability to hide their presence for long. Lycosid collects the information from the guest OS without relying on the fact that VMM implements each and every detail on guest OS. The shortcoming is that it cannot obtain the correct process list when the system is running, since the process list is changing every moment (*i.e.*, time synchronization). The key lies in taking a trusted view of the operating system generating implicit view that carries a greater level of details. Dunlap *et al.* [41] presented a prototype ReVirt. The problem addressed is that the log files might be tampered or completely removed by a skilled attacker. Thus leaving no way to detect what happened before and during the system compromise. ReVirt is an architecture that stores the log files out of the VM so that any tampering done in the VM does not affect the logs.

DKSM attack [8], demonstrated through a prototype implementation, tends to subvert introspection, *i.e.*, out-of-box intrusion detection. The scheme addresses a very important assumption on which the introspection techniques, namely Livewire and VMwatcher, work. As mentioned above, the potential semantic gap is bridged by the intrusion detection techniques deployed in Livewire, VMwatcher, *etc.* These techniques rely heavily on the template to gain a low-level view of the guest OS. The low-level view incorporated in the data structures reveal the VM processes' and files' details. These data

structures make use of kernel data. In short, using the templates, the guest OS is relying on the usage of kernel data in a specific manner. If the kernel is compromised, or the template tampered, then the assumption that a guest OS is respecting its own kernel fails and the low-level view thus acquired will be incorrect. However, this attack is detectable in most Windows and Linux operating systems [42].

In Table 4, from characteristics and limitations, it can be concluded that underlying infrastructure (*e.g.* hypervisor) security is not covered by IDS. Furthermore, an attack launched on the hypervisor will have the capability to compromise VMs running on it.

Table 4. Analysis of VMM in cloud.

Research	Deployment	Technique	Characteristic	Limitation
[29]	Outside MVM (type I)	Signature and Anomaly based Detection (not complex detection)	Basis for all other VMI	Not real time
[35]	Outside MVM (type II)	View-comparison based malware detection	Semantic reconstruction, live monitoring without affecting VMM, support for multiple VMM	Does not mitigate zero-day threats [26], attacks on hypervisor can compromise approach [26], timing problem in reconstructing views
[23]	Outside MVM (type II)	Anomaly based Detection	System activities (system calls *etc.*), Network traffic.	Monitoring code inside VMM, VMM code modified
[36]	PVM	Security VM depends on API (anti-virus) used by end user	Hooks and trampoline function inside VMM	VMM Code modified, hooks and trampoline bottleneck [18]
[41]	Inside/outside VMM	Analyzing logs	Secure logging	No malware detection [26], time to analyze, record and replay logs
[39]	VMM	Anomaly based Detection	Detects hyper-call based attacks targeting the VMM	Cannot detect other attacks [22]
[40]	VMM	View comparison based malware detection	Can detect and identify rootkits efficiently	Timing problem in reconstructing views, cannot detect idle hidden process, only for rootkits

3.4 Distributed Intrusion Detection Systems

To counter DoS and DDoS attacks in cloud, Lo *et al.* [43] have proposed and simulated an intrusion detection system. The IDS has four components each playing a specific role. This protects the system from a single point of failure. However, it uses signature-based detection technique due to which unknown attacks are not detected. He *et al.* [44] propose a 3-D IDS architecture. It is a distributed IDS for the users of IaaS in cloud. 3-D IDS is composed of a server and multiple agents. The architecture is a theoretical model and no experimental evidence is presented in the paper. Moreover, the architecture requires the deployment of the server at user end which is not always necessarily deployed at all users.

Shelke *et al.* [45] propose a solution for Cross Site Scripting (XXS) and DDoS attacks using a multi-threaded network intrusion detection system. The method consists of three modules namely capture module, analysis and processing module, and finally the reporting module. It is a novel approach but the researcher has not provided evidence to

prove the concept. Siren [46] is another VM-based intrusion detection system. Siren works by detecting malicious software running in the VM that attempt to send information over the network of which the VM is part of. Siren works on the principle that there should be no traffic generated by VM on the network in the absence of human input. If such a state is detected, then Siren flags the traffic as malicious. One of the best features of Siren is its ability to inject crafted human input to investigate for the ad-on malware. The technique seems promising however the real challenge lies with generating traffic that resembles closely with the human input.

Dastjerdi *et al.* [47] propose another DIDS based on static and mobile agents. They have modified the DIDMA [48] approach for a DIDS to work in the virtual cloud environments. Dastjerdi *et al.* mention that their technique reduces network load for less than six VMs in a network. Each mobile agent only analyses a small amount of code. However, if this limit is exceeded, the network loads gets heavier [13, 14]. Roschke *et al.* in [34] propose IDS which they deploy on each virtual machine. They mention that this IDS can be either HIDS or NIDS based on the type of sensors deployed and hence the type of data monitored. The drawback of this technique is that it uses multiple IDS sensors and each sensor is deployed on the virtual machine. Ibrahim *et al.* [42] propose a technique that uses VMware specially built APIs to carry out VM introspection. Their architecture has two main modules VMI back-end and CloudSec. The authors claim that their technique supports real-time monitoring while bridging the semantic gap. They tested their prototype and found minor overheads.

The characteristics and limitations highlighted in Table 5 help us to reach the conclusion that the distributed IDS limits the number of VMs. Furthermore, attention should be provided to reduce computational overhead.

Table 5. Analysis of DIDS in cloud.

Research	Deployment	Technique	Characteristic	Limitation
[47]	MVM	Anomaly and Signature based Detection	Works even if VM migrated, can detect known and zero-day attacks	Limited number of VM can be visited, performance overhead for malicious MA
[42]	One module inside VMM, one external	Anomaly based Detection	Low overheads, semantic reconstruction	Execution suspended in case of event, no defense mechanism, rootkits only
[44]	Major module (agent) is deployed inside VM	Signature, anomaly, vertical & horizontal fusion analysis	Integrates HIDS, NIDS techniques, correlates VM data	No prototype
[43]	Every single cloud region	Signature based Detection	Can detect from signatures of known attacks	High computation overhead and unable to detect unknown attacks
[45]	Processing Server	Hybrid	Signature based, log files for auditing, user profiles.	No implementation

4. PERFORMANCE ANALYSIS OF EXISTING CLOUD BASED IDS

Different approaches for cloud intrusion detection have been discussed in the above section, each one addressing somewhat unique research gaps. However, each technique has its own strengths and limitations as well. A HIDS deployed on a virtual machine comes with the inherent flaw of low attack resistance [23]. It provides good visibility into the system, however, a HIDS on the host system does not cater virtual machines. It

would only act as traditional HIDS monitoring the host system itself. Such a technique in conjunction with other techniques (*e.g.*, VMI-based) can prove to be efficient in securing the host system itself.

A NIDS can be deployed on each virtual machine. However, the most widely used technique is that of deploying a NIDS on a virtual switch. Such a technique costs a lot of computational overhead as one component has to cater the burden of all traffic and segregate it as well. In high-traffic environments, the IDS might fail due to this complexity and render all detection results unreliable. If the only route for all traffic gets suspended due to this reason, it may even result in a severe DoS attack until the NIDS is restarted [29]. Additionally, a NIDS has low visibility into the VM and would not be able to detect attacks occurring internally inside the hypervisor [13]. A NIDS is incapable of analyzing any encrypted traffic as well. However, on the brighter side a NIDS has high attack resistance [23].

VMI or VMM based techniques assume that the hypervisor remains secure and non-malicious. However, this is a widely accepted assumption as the code for VMM is small and hence less prone to contain bugs [38]. Modi *et al.* in [13] mention that generally a vSwitch and hypervisor are part of the Trusted Cloud Base. The reasons that they mention are as following:

1. The code is small and less prone to bugs.
2. Their security can be strengthened further by use of a Trusted Platform Module (TPM).
3. They are under full control of the cloud service provider.

Wang *et al.* in [49], however, argue that many VMMs do have a large code base so this is not always an applicable assumption. They state that from 2007 to 2010, National Vulnerability Database (NVD) recorded 26 vulnerabilities in Xen Hypervisor and 18 in VMware.

When compared with NIDS, VMI techniques provide better visibility into the system and solve the problem of attacker manipulation as well. As long as the kernel data structures are intact, a hypervisor-level IDS will continue its operation in a reliable manner even if the guest OS has been compromised. The performance of VMI/VMM based techniques also depend on the fact that the VMI remains hidden from the attacker. Garfinkel *et al.* in [29] mention that it is almost impossible to hide the presence of a VMM due to the difference in execution of instructions and operations. This is a performance reduction factor for VMI techniques in general.

The problem of semantic gap has also been discussed in the literature [50]. From outside the guest OS, a VMI technique gathers the hardware level view of the VM. This means decreasing the visibility as compared to in-guest IDS. Using knowledge about the kernel data structures and OS algorithms, these techniques attempt to build a high-level view of the system. However, this means bringing some pause in detection.

In other solutions, where some monitoring code is deployed inside the guest OS, real-time monitoring is provided. The monitoring code helps gather high-level view of the virtual machine. However, this technique brings with its own limitations. The presence of monitoring code can affect the deployed code and make it behave differently [50]. Besides, if it is deployed inside the guest OS, it could mean that the code gets operational only after the guest OS is booted and gets switched off before the guest OS has completely shut down [50]. The monitoring code also increases the computational load

of the guest OS itself. Additionally, the guest OS needs to be suspended for analysis when this technique is used [29, 42].

For CloudSec implementation, Ibrahim *et al.* [42] claim that their technique supports real-time monitoring while bridging the semantic gap. Some instruction suspension is however still present in this technique. When the back-end module passes any memory page to the Semantic Gap Bridge (SGB) for analysis, VM operations are suspended to keep the monitoring real-time.

VMI techniques are very complex in nature. Detailed and up-to-date information about the internal workings of the specific operating system is required. Although this is a complicated process even for an open-source system, closed-source systems would require a considerable amount of reverse engineering as well [51]. Moreover, any system update or installed patch would mean that the complete data gets changes hence leaving the introspection tool invalid [51].

Dolan-Gavitt *et al.* in [51] present "Virtuoso" a technique to automatically generate introspection tool in three phases. In the first (training) phase, a code snippet inside the operating system records all OS related information to be used by the tool. The second phase (analysis) extracts security related information for this or information specifically needed for introspection. The information is then used to generate an introspection program that can run outside the guest OS, specifically a runtime environment provided by phase 3. The timing problem in reconstructing semantic views is mentioned as a limitation for the technique given in [35]. This is because the IDS uses comparison-based approach to detect attacks. There would always be some difference between views taken from guest OS and VMM if they are not exactly synchronized. This can lead to lots of false negatives [52]. When VMM code is modified, it is mentioned as a limitation because all VMI-based techniques depend on the assumption that VMM is from the trusted code base. Any changes to its code can lead to significant bugs and hence pave the way for possible attacks.

Payne *et al.* argue in [36] that the hook and the trampoline function are security bottlenecks [50]. Although the authors state that the code for these functions is very small and contained, and hence reliable, it is evident that the code still resides inside the monitored VM and hence is subject to attacks. In [41], the authors use logging functionality as the basis for detecting attacks. The tool can be deployed inside or outside the VM. If deployed inside the VM, it would not be resistant to attacks. The timing overhead incurred during recording, replaying, and analyzing the attacks is a limitation. The malicious entity can cause significant damage before the IDS is able to detect an attack.

The DIDS technique used in [47] mentions virtualization as a way of dealing with malicious mobile agents. However, for the virtual cloud, mobile agents are deployed inside monitored VM. The performance impact that would occur as a result of shifting their position to another VM has not been discussed. It could pose significant overheads and has hence been mentioned as a limitation.

Patel *et al.* [11] mention eight performance requirements for any IDS techniques generally used in the cloud. We have used six of these requirements as another way of evaluating the techniques surveyed in this paper. The analysis is presented in Table 7. A noteworthy point is that the evaluation here is a comparison based, *i.e.*, during analysis a technique is analyzed whether it is fast when compared to other techniques. A brief description is provided in Table 6.

Table 6. Performance requirements for any IDS technique [11].

Criteria	Description
Detect in large, distributed, and multi-tiered environment (R1)	The nodes are created, updated, and deleted at runtime in the cloud. IDS should be capable of managing with minimum or no human interaction.
Detect variety of attacks (R2)	Wherever signature based techniques are used, detecting a variety of attacks is not possible, hence, the field has been marked in negative.
Fast Detection (R3)	NIDS techniques are not fast in general since they have to analyze data from a lot of sources. The field has been marked in negative accordingly.
Self-Adoption (R4)	It should automatically adapt to the changes in cloud environment, *i.e.*, node addition or removal.
Scalable (R5)	It means whether the technique can be tailored to be used for large systems or not. Techniques which cannot support large virtual networks have been marked negative.
Real Time Monitoring (R6)	VMI-based techniques are in general not real-time. However, some of the techniques mentioned in this paper have specifically been constructed as an attempt to fill this gap.
Synchronization issues (R7)	Wherever IDS is composed of multiple components or depends on real-time communication with any other component, synchronization can be a problem.
Resistant to compromise (R8)	Wherever the IDS depends on some component inside the VM, the technique becomes less attack resistant.
IDS works in SDN-based cloud environment (R9)	SDN consists of a centralized controller and dumb switches. We evaluate whether the IDS schemes proposed for traditional networking can be deployed in SDN as well.

Table 7. Performance analysis of existing techniques.

Research	Detect Variety of Attacks	Fast Detection	Scalable	Real Time Monitoring	Synchronization Issues	Resistant to Compromise	Works in SDN-based Cloud
Roschke, *et al.* [34]	✓	×	×	✓	✓	×	N/A
Bakshi *et al.* [30]	×	×	×	✓	×	×	✓
Garfinkel *et al.* [29]	✓	N/A	N/A	×	×	N/A	✓
Jiang *et al.* [35]	✓	N/A	N/A	N/A	×	N/A	N/A
Laureano *et al.* [23]	✓	N/A	N/A	×	×	N/A	N/A
Payne *et al.* [36]	N/A	N/A	N/A	✓	×	N/A	N/A
Gupta *et al.* [33]	✓	×	N/A	✓	✓	✓	N/A
Dunlap *et al.* [41]	✓	N/A	N/A	N/A	N/A	N/A	N/A
Bharadwaja *et al.* [39]	×	N/A	N/A	✓	N/A	N/A	×
Jones *et al.* [40]		N/A	N/A	×	N/A	N/A	N/A
Dastjerdi *et al.* [47]	N/A	N/A	✓	✓	✓	N/A	×
Ibrahim *et al.* [42]	×	×	N/A	✓	N/A	×	×
He *et al.* [44]	✓	✓	✓	✓	✓	×	×
Lo *et al.* [43]	×	✓	✓	✓	✓	×	×
Mazzariello *et al.* [32]	×	×	N/A	✓	×	×	×
Shelke *et al.* [45]	✓	✓	✓	✓	✓	×	×
Patel *et al.* [25]	✓	✓	✓	✓	×	✓	N/A
Lee *et al.* [26]	✓	×	N/A	✓	N/A	✓	N/A
Vieira *et al.* [27]	✓	×	✓	✓	×	✓	N/A
Dhage *et al.* [28]	✓	N/A	✓	×	N/A	✓	N/A

In this paper we have not evaluated the IDSs' performance on the criteria R1 and R4, given in Table 6, since authors discussing their techniques have not provided sufficient information to evaluate them on the given parameters.

In "**Works in SDN-based Cloud environment**" (R9) the requirement not applicable (N/A) is selected for some techniques because the authors do not take responsibility for the underlying technology. They only provide IDS for the VM. In SDN network based DIDS techniques cannot be implemented as all traffic is monitored by the controller.

5. COMMON CHALLENGES AND OPEN ISSUES OF IDS

With the evolution of networks and computer infrastructure, intrusion detection techniques have improved with the goal of providing security and protection. Despite extensive research in this area, there are still open problems. Some of the most significant challenges of implementing an intrusion detection system include low detection efficiency, low throughput & high cost IDS, lack of standard metrics & assessment methodologies and encrypted data.

High false-positive rate results in low detection efficiency [53]. In anomaly based IDS less training time leads to more false positives while more training time results in more resource utilization. A balance between the two factors is required *i.e.* security and usability. High data rates (Gbps) in wideband technologies lead to low throughput and high cost IDS [54]. To overcome this problem grid computing based and distributed detection techniques are proposed. Selection of IDS is a difficult process due to lack of standard metrics and assessment methodologies [55, 56]. According to Axelsson *et al.* report, IDS itself are attacked and no countermeasures for their protection are placed [57]. Ptacek *et al.* propose different mechanisms to secure the IDS [58]. One of the most significant hurdle faced by IDS in every platform is encrypted data. The above-mentioned points should be catered for in designing and implementing an IDS.

5.1 Lack of Datasets for Cloud IDS

In recent years the attacks against cloud computing have evolved and it is observed that the lack of datasets hinders the implementation of an effective intrusion detection system. The current datasets used for traditional computing cannot be used due to the heterogeneous operating systems installed in virtual machines, diversity in user requirements, and the data size of cloud.

Kholidy *et al.* [59] propose a cloud intrusion detection dataset (CIDD), the only dataset designed specifically keeping the infrastructure of cloud in mind. It consists of knowledge and user based audit data collected from Unix and Window users. CIDD includes audit parameters which can detect host based, network based, and masquerade attacks. However, the dataset still lacks sufficient amount of data for a wider detection.

The effectiveness of an intrusion detection dataset is determined by the true and false positive rate. The true positive rate is determined by sending attacks to the cloud IDS and evaluating the number of attacks detected. False positive rate determines the ratio of false alert. The false positive rate of an ideal IDS will be zero, *i.e.*, no false alert

is generated. Building a cloud dataset is challenging due to several major reasons highlighted below:

1. Real-life attacks data is not available for researching solutions and models. After an attack data is labeled as evidence and not made public for examination by researchers.
2. The infrastructure of a commercial cloud is difficult to be controlled by a researcher to build attack scenario. Furthermore, private cloud vulnerabilities are similar to conventional IT infrastructure which adds to the hindrance in building attack scenarios [60].
3. The diversity of operating systems in VMs (*e.g.* Unix, Windows) makes it difficult to collect data from different users.
4. The enormous size of audit data and a high number of users in cloud computing require great computing resources.

5.2 How to Detect Application Level DDoS Attacks in Cloud?

Application level DDoS flooding attack is one of the most dangerous types of DDoS attacks because they comparatively utilize less bandwidth and are stealthier than volumetric attack. The application level DDoS flooding attack effects the services in a similar manner as volumetric attack, since they target specific characteristics of an application, *e.g.* HTTP, DNS.

Research by Gartner shows an increase in the incidents initiated by application level DDoS flooding attacks [61]; and for their mitigation, access to information in the payload is required. Currently, IDS in cloud are capable enough to detect application layer attack, however, they cannot detect DDoS attacks which use valid packets [62]. Most of the attackers today use valid packets. To some extent anomaly based IDS offer capabilities to detect these attacks but an expert needs to manually tune it, still IDS might not be able to detect all the attack flows. Additionally, IDS only detects and generates an alarm. Using IDS as a DDoS defense platform raises a lot of issues as they will not perform any action to mitigate the threat [63]. A complementary mitigation strategy is required in IDS which will detect extremely sophisticated attack flow and take necessary steps. Signature based IDS will miss application layer DDoS attack. Sophisticated DDoS attacks are identified by anomalous behavior at L3 and L4, and IDS is not optimized enough to detect and mitigate DDoS.

Major efforts are essential to propose a solution which is a perfect trade-off between performance and security.

5.3 Securing SDN

Software defined networking is adopted in cloud computing because it is programmable, easily partitionable, and virtualized. In terms of identification and response of attacks, SDN has two key advantages over traditional networks [64]: 1) the control plane allows an administrator to separate and block attack patterns simultaneously on all heterogeneous hardware (no need to individually reconfigure them), and 2) instead of investing in an expensive intrusion detection system, SDN can make it a distributed task among nodes, *e.g.* controller can define rules on switches to detect malicious flows.

This also gives an opportunity to the attackers as SDN exposes new interfaces, *i.e.*,

communication between the control plane and data plane. By compromising the SDN controller, the whole network can be compromised. Moreover, low-level network services can also be attacked which was not possible in traditional networks. Therefore, while using SDN to provide IDS services, the security of SDN should be kept in mind. An airtight access control policy for the SDN controller is needed to be designed and implemented.

5.4 Secure Hypervisor

To the best of our knowledge, current literature on secure cloud lacks discussion on capabilities, limitations, and applicability of secure hypervisor. The security of cloud greatly depends on the hypervisor on which the virtual machines are hosted. A single hypervisor can host many VM at the same time. A scenario in which VM running on a hypervisor running multiple VMs get compromised. There is no guarantee that the intruder would not be able to access the hypervisor and ultimately all the VMs running on it. A perfectly secured VM can be violated by a compromised hypervisor. Therefore, at the time of service level agreement, ask the hard questions; How are VMs isolated? How is a security issue in hypervisor detected? What will be the response of the cloud provider?

The examples of a secure hypervisor are sHype and NoHype implementation. The sHype is an IBM initiated project which was initially developed for rHype, an open-source research hypervisor, however, it has been implemented in Xen which is an open-source hypervisor as well. The fundamental goal of sHype project was to control inter VMs information flow. The architecture of sHype is very flexible and supports a wide range of security policies. Nevertheless, it does not control all the information flows between the VMs, just explicit data is monitored [65]. In NoHype [66], researchers have removed virtualization but kept the key features of virtualization. This secures the VMs from attacks through a compromised hypervisor due to a malicious VM. The limitations of this technique are one core per VM, memory space partitioning, and virtualization of I/O devices at hardware level [66]. They basically limit the covert channel by one VM per core but research shows that there are other covert channels in the architecture [67]. Thus this is a weak technique.

The use of these techniques for the purpose of securing virtual cloud in general, and VMI-IDS in particular can be considered in future works.

5.5 VM Migration

Current literature is limited on the possible impact of virtual machine migration on intrusion detection system in the cloud. Attackers can gain full control of a VM in the migration process through stack, heap or integer overflow vulnerability in the migration module. Additionally, without proper policies, an intruder can initiate or terminate the VM migration process which may result in denial of service, injection of malicious code during migration or gaining control of the VM during migration. Moreover, insecure channels used during migration also give the attacker a window to launch passive and active attacks [68].

Ahmad *et al.* [68] have conducted a survey on different techniques that secure VM

migration process. For this purpose, they have highlighted the vulnerabilities, threats, and different possible attacks in the VM migration process. Moreover, they have identified security requirements for secure VM migration, and evaluated the existing mechanisms. VM migration can have a significant impact on intrusion detection policies. The technique proposed by Dastjerdi *et al.* [47] supports VM migration but the latter's effects on other techniques have not been considered in this work. Future work within the same domain can target filling this gap as well.

5.6 Effective and Efficient IDS in Cloud Architecture

In the context of cloud security, the role of intrusion detection systems is vital. As discussed in section 3, a significant number of detection systems have been proposed, however, they are unable to provide complete security. To some extent hybrid detection technique disturbed intrusion detection system provides a secure environment but it requires a trade-off with the performance. For the betterment of detection system, an algorithm is required which uses minimal computing resources and provides a secure environment. Moreover, the classifier and feature selection should also detect resource leakage [69].

6. CONCLUSION

The confidentiality, integrity, and availability of a computer system is ensured using IDS. Due to the exponential growth of cloud users, IDSs for cloud computing are in great demand. In this paper, we have discussed existing solutions for intrusion detection in the cloud. Cloud based IDS have been divided into four types, including: network-based, host-based, distribution based, and virtual machine introspection based systems. Their limitations and unique capabilities are mentioned and mapped on to the performance commensurate with that required for a cloud-based IDS in general. Performance criteria used for evaluation is derived from the basic characteristics of a cloud and an additional requirement has been proposed, *i.e.*, effective working of an intrusion detection technique in a SDN-based cloud environment. SDN capabilities include software based traffic analysis, global view of network, and a centralized control. However, SDN has its own set of security issues to the table.

In addition to this, common pitfalls in the implementation of an IDS have been discussed. The significant issues, including lack of datasets to evaluate the performance of an IDS in a cloud, how to detect application level DDoS attacks in a cloud, securing SDN, secure hypervisor, VM migration, and effective and efficient IDS in a cloud architecture have been highlighted. In summary, a lot of efforts have been made towards securing cloud using IDS, however, due to the evolving nature of cloud, *i.e.*, scalability, distributed processing, big data analysis, and service oriented architecture, there is still room of improvement to achieve the ideal IDS.

REFERENCES

1. “Google apps for work – Gmail, drive, docs and more,” Google.com, 2016, http://www.google.com/apps/business.

2. "Google apps engine," Code.google.com, 2016, http://code.google.com/appengine.
3. "Microsoft azure: Cloud computing platform & services," Microsoft.com, 2016, http://www.microsoft.com/azure.
4. "Amazon Web Services (AWS) – Cloud computing services," Amazon Web Services, Inc., 2016, http://aws.amazon.com.
5. "Eucalyptus," 2016, http://eucalyptus.cs.ucsb.edu/.
6. "OpenNebula – Flexible enterprise cloud made simple," Opennebula.org, 2016, http://www.opennebula.org.
7. P. Mell and T. Grance, "The NIST definition of cloud computing," http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf, 2011.
8. S. Bahram, X. Jiang, Z. Wang, M. Grace, J. Li, D. Srinivasan, *et al.*, "DKSM: Subverting virtual machine introspection for fun and profit," in *Proceedings of the 29th IEEE Symposium on Reliable Distributed Systems*, 2010, pp. 82-91.
9. S. T. King, P. M. Chen, Y.-M. Wang, C. Verbowski, H. J. Wang, and J. R. Lorch, "SubVirt-Implementing malware with virtual machines," in *Proceedings of IEEE Symposium on Security and Privacy*, 2006, pp. 314-327.
10. J. Rutkowska, "Subverting VistaTM kernel for fun and profit," presented at the *Black Hat Conference*, 2006.
11. A. Patel, M. Taghavi, K. Bakhtiyari, and J. C. Júnior, "An intrusion detection and prevention system in cloud computing: A systematic review," *Journal of Network and Computer Applications*, Vol. 36, 2013, pp. 25-41.
12. M. Zbakh, K. Elmahdi, R. Cherkaoui, and S. Enniari, "A multi-criteria analysis of intrusion detection architectures in cloud environments," in *Proceedings of International Conference on Cloud Technologies and Applications*, 2015, pp. 1-9.
13. C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," *Journal of Network and Computer Applications*, Vol. 36, 2013, pp. 42-57.
14. Y. Mehmood, M. A. Shibli, U. Habiba, and R. Masood, "Intrusion detection system in cloud computing-challenges and opportunities," in *Proceedings of the 2nd National Conference on Information Assurance*, 2013, pp. 59-66.
15. U. Oktay and O. K. Sahingoz, "Attack types and intrusion detection systems," in *Proceedings of the 6th International Information Security and Cryptology Conference*, 2013, pp. 71-76.
16. N. A. Prematilaka, A. C. Aponso, and N. Krishnarajah, "Review on state of art intrusion detection systems designed for the cloud computing paradigm," in *Proceedings of the 47th International Carnahan Conference on Security Technology*, 2013, pp. 1-6.
17. M. H. Sqalli, F. Al-Haidari, and K. Salah, "EDoS-Shield − A two-steps mitigation technique against EDoS attacks in cloud computing," in *Proceedings of the 4th IEEE International Conference on Utility and Cloud Computing*, 2011, pp. 49-56.
18. M. D. Gaithersburg, "Invulnerability requirements for cryptographic modules," in National Institute of Standards and Technology, ed., Federal Information Processing Standards 140-2, 2001, May 25.
19. W. Voorsluys, J. Broberg, and R. Buyya, "Introduction to cloud computing," *Cloud Computing: Principles and Paradigms*, 2011, pp. 1-44.
20. A. Mohta, R. K. Sahu, and L. K. Awasthi, "Robust data security for cloud while

using third party auditor," *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 2, 2012.
21. M. Slaviero, "Black Hat presentation demo vids," http://www.sensepost.com/blog/3797.html.
22. J. C. Roberts II and W. Al-Hamdani, "Who can you trust in the cloud?: a review of security issues within cloud computing," in *Proceedings of Information Security Curriculum Development Conference*, 2011, pp. 15-19.
23. M. Laureano, C. Maziero, and E. Jamhour, "Intrusion detection in virtual machine environments," in *Proceedings of the 30th Euromicro Conference*, 2004, pp. 520-525.
24. S. Alarifi and S. Wolthusen, "Anomaly detection for ephemeral cloud IaaS virtual machines," *Network and System Security*, 2013, pp. 321-335.
25. A. Patel, Q. Qassim, Z. Shukor, J. Nogueira, J. Júnior, C. Wills, *et al.*, "Autonomic agent-based self-managed intrusion detection and prevention system," in *Proceedings of the South African Information Security Multi-Conference*, 2011 pp. 223-234.
26. J.-H. Lee, M.-W. Park, J.-H. Eom, and T.-M. Chung, "Multi-level intrusion detection system and log management in cloud computing," in *Proceedings of the 13th International Conference Advanced Communication Technology*, 2011 pp. 552-555.
27. K. Vieira, A. Schulter, C. Westphall, and C. Westphall, "Intrusion detection for grid and cloud computing," *IT Professional Magazine*, Vol. 12, 2010, p. 38.
28. S. N. Dhage and B. Meshram, "Intrusion detection system in cloud computing environment," *International Journal of Cloud Computing*, Vol. 1, 2012, pp. 261-282.
29. T. Garfinkel and M. Rosenblum, "A virtual machine introspection based architecture for intrusion detection," in *Proceedings of the 2nd International Conference on Communication Software and Networks*, Vol. 3, 2003, pp. 191-206.
30. A. Bakshi and Y. B. Dujodwala, "Securing cloud from DDOS attacks using intrusion detection system in virtual machine," in *Proceedings of the 2nd International Conference on Communication Software and Networks*, 2010, pp. 260-264.
31. "Snort – Network intrusion detection & prevention system," Snort.org, 2016, http://www.snort.org/.
32. C. Mazzariello, R. Bifulco, and R. Canonico, "Integrating a network IDS into an open source Cloud Computing environment," in *Proceedings of the 6th International Conference on Information Assurance and Security*, 2010, pp. 265-270.
33. S. Gupta, P. Kumar, and A. Abraham, "A profile based network intrusion detection and prevention system for securing cloud environment," *International Journal of Distributed Sensor Networks*, Vol. 2013, 2013, pp. 1-12.
34. S. Roschke, F. Cheng, and C. Meinel, "An extensible and virtualization-compatible IDS management architecture," in *Proceedings of the 5th International Conference on Information Assurance and Security*, Vol. 2, 2009 pp. 130-134.
35. X. Jiang, X. Wang, and D. Xu, "Stealthy malware detection through vmm-based out-of-the-box semantic view reconstruction," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, 2007, pp. 128-138.
36. B. D. Payne, M. Carbone, M. Sharif, and W. Lee, "Lares: An architecture for secure active monitoring using virtualization," in *Proceedings of IEEE Symposium on Security and Privacy*, 2008, pp. 233-247.
37. B. D. Payne, M. D. P. D. A. Carbone, and W. Lee, "Secure and flexible monitoring of virtual machines," in *Proceedings of the 23rd Annual Computer Security Applica-*

tions Conference, 2007, pp. 385-397.
38. U. Tupakula, V. Varadharajan, and D. Dutta, "Intrusion detection techniques for virtual domains," in *Proceedings of the 19th International Conference on High Performance Computing*, 2012, pp. 1-9.
39. S. Bharadwaja, W. Sun, M. Niamat, and F. Shen, "Collabra: A Xen hypervisor based collaborative intrusion detection system," in *Proceedings of the 8th International Conference on Information Technology: New Generations*, 2011, pp. 695-700.
40. S. T. Jones, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "VMM-based hidden process detection and identification using Lycosid," in *Proceedings of the 4th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, 2008, pp. 91-100.
41. G. W. Dunlap, S. T. King, S. Cinar, M. A. Basrai, and P. M. Chen, "ReVirtenabling intrusion analysis through virtual-machine logging and replay," *ACM SIGOPS Operating Systems Review*, Vol. 36, 2002, pp. 211-224.
42. A. S. Ibrahim, J. Hamlyn-Harris, J. Grundy, and M. Almorsy, "CloudSec – a security monitoring appliance for Virtual Machines in the IaaS cloud model," in *Proceedings of the 5th International Conference on Network and System Security*, 2011, pp. 113-120.
43. C.-C. Lo, C.-C. Huang, and J. Ku, "A cooperative intrusion detection system framework for cloud computing networks," in *Proceedings of the 39th International Conference on Parallel Processing Workshops*, 2010, pp. 280-284.
44. J. He, C. Tang, Y. Yang, Y. Qiao, and C. Liu, "3D-IDS: IaaS User-oriented intrusion detection system," in *Proceedings of International Symposium on Information Science and Engineering*, 2012, pp. 12-15.
45. M. P. K. Shelke, M. S. Sontakke, and A. D. Gawande, "Intrusion detection system for cloud computing," *International Journal of Scientific and Technology Research*, 2012, pp. 67-71.
46. X. Zhao, B. Kevin, and P. Atul, "Virtual machine security systems," *Advances in Computer Science and Engineering*, 2009, pp. 339-365.
47. A. V. Dastjerdi, K. A. Bakar, and S. G. H. Tabatabaei, "Distributed intrusion detection in clouds using mobile agents," in *Proceedings of the 3rd International Conference on Advanced Engineering Computing and Applications in Sciences*, 2009, pp. 175-180.
48. P. Kannadiga and M. Zulkernine, "DIDMA – A distributed intrusion detection system using mobile agents.pdf," in *Proceedings of the 6th International Conference on Software Engineering*, *Artificial Intelligence*, *Networking and Parallel/Distributed Computing and the 1st ACIS International Workshop on Self-Assembling Wireless Network*, 2005, pp. 238-245.
49. Z. Wang and X. Jiang, "HyperSafe: A lightweight approach to provide lifetime hypervisor control-flow integrity," in *Proceedings of IEEE Symposium on Security and Privacy*, 2010, pp. 380-395.
50. A. More and S. Tapaswi, "Virtual machine introspection – towards bridging the semantic gap," *Journal of Cloud Computing Advances*, *Systems and Applications*, Vol. 3, 2014, p. 1.
51. B. Dolan-Gavitt, T. Leek, M. Zhivich, J. Giffin, and W. Lee, "Virtuoso: Narrowing the semantic gap in virtual machine introspection," in *Proceedings of IEEE Sympo-*

sium on Security and Privacy, 2011, pp. 297-312.
52. R. Denz and S. Taylor, "A survey on securing the virtual cloud," *Journal of Cloud Computing: Advances*, *Systems and Applications*, Vol. 2, 2013, p. 17.
53. S. Axelsson, "The base-rate fallacy and its implications for the difficulty of intrusion detection," in *Proceedings of the 6th ACM Conference on Computer and Communications Security*, 1999, pp. 1-7.
54. C. Kruegel, F. Valeur, G. Vigna, and R. Kemmerer, "Stateful intrusion detection for high-speed network's," in *Proceedings of IEEE Symposium on Security and Privacy*, 2002, pp. 285-293.
55. J. E. Gaffney and J. W. Ulvila, "Evaluation of intrusion detector; A decision theory approach," in *Proceedings of IEEE Symposium on Security and Privacy*, 2001, pp. 50-61.
56. S. J. Stolfo, W. L. Wei Fan, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection; Results from the JAM project," in *Proceedings of DARPA Information Survivability Conference and Exposition*, Vol. 2, 2000, pp. 130-144.
57. S. Axelsson, "Research in intrusion-detection systems: a survey," Technical Report 98-171998, Department of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden.
58. T. H. Ptacek and T. N. Newsham, "Insertion, evasion, and denial of service: Eluding network intrusion detection," DTIC Document, 1998.
59. H. A. Kholidy and F. Baiardi, "CIDD: A cloud intrusion detection dataset for cloud computing and masquerade attacks," in *Proceedings of the 9th International Conference on Information Technology: New Generations*, 2012, pp. 397-402.
60. M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges," *Information Sciences*, Vol. 305, 2015, pp. 357-383.
61. D. Sher, "Gartner: Application layer DDoS attacks to increase in 2013," Neptune Web, Inc., https://www.corero.com/blog/361-gartner-application-layerddos-attacks-to-increase-in-2013.html, 2016.
62. Riverhead WP, http://www.cse.msu.edu/~cse825/Riverhead_WP.pdf, 2016.
63. A. Carlin, M. Hammoudeh, and O. Aldabbas, "Intrusion detection and countermeasure of virtual cloud systems: State of the art and current challenges," *International Journal of Advanced Computer Science and Applications*, Vol. 6, 2015, pp. 1-15.
64. M. Tsugawa, A. Matsunaga, and J. A. B. Fortes, "Cloud computing security: What changes with software-defined networking?" in *Secure Cloud Computing*, S. Jajodia, K. Kant, P. Samarati, A. Singhal, V. Swarup, and C. Wang, eds., Springer, NY, 2014, pp. 77-93.
65. S. Vogl, "Secure hypervisors," in *Proceedings of the 12th International Conference on Enterprise Information System*, 2010, pp. 1-16.
66. E. Keller, J. Szefer, J. Rexford, and R. B. Lee, "NoHype – virtualized cloud infrastructure without the virtualization," in *Proceedings of ACM SIGARCH Computer Architecture News*, Vol. 38, 2010, pp. 350-361.
67. E. Caron, F. Desprez, and J. Rouzaud-Cornabas, "Smart resource allocation to improve cloud security," in *Security*, *Privacy and Trust in Cloud Systems*, S. Nepal and M. Pathan, ed., Springer, Berlin Heidelberg, 2014, pp. 103-143.
68. N. Ahmad, A. Kanwal, and M. A. Shibli, "Survey on secure live virtual machine

(VM) migration in cloud," in *Proceedings of the 2nd National Conference on Information Assurance*, 2013, pp. 101-106.
69. C. Murthy, A. S. Manjunatha, A. Jaiswal, and B. R. Madhu, "Building efficient classifiers for intrusion detection with reduction of features," *International Journal of Applied Engineering Research*, Vol. 11, 2016, pp. 4590-4596.

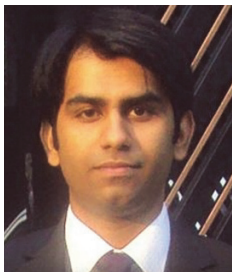
Amna Riaz is an MS Information Security student at School of Electrical Engineering and Computer Science, a campus of National University of Sciences and Technology (NUST), Islamabad, Pakistan. She has completed her Bachelors of Engineering in Electrical (Telecom.) domain from Military College of Signals, NUST. Her research interests include network security, cloud computing security, digital forensics and cryptography.

Hafiz Farooq Ahmad is working as an Associate Professor at the College of Computer Sciences and Information Technology, King Faisal University, Alahsa, Saudi Arabia. He has worked previously at Comtec, Japan, DTS Japan and SEECS, NUST Pakistan. He holds Ph.D. (Computer Science) from Tokyo Institute of Technology, Tokyo, Japan. His research interests are in distributed computing including semantic systems, health informatics and web application security.

Junaid Qadir is an Associate Professor at the Information Technology University (ITU)-Punjab, Lahore, Pakistan. He is the Director of the IHSAN Lab at ITU that focuses on deploying ICT for development, and is engaged in systems and networking research. Prior to joining ITU, he was an Assistant Professor at the School of Electrical Engineering and Computer Sciences (SEECS), National University of Sciences and Technology (NUST), Pakistan. At SEECS, he directed the Cognet Lab at SEECS that focused on cognitive networking and the application of computational intelligence techniques in networking. He has been awarded the highest national teaching award in Pakistan – the higher education commission's (HEC) best university teacher award for the year 2012-2013. His research interests include the application of algorithmic, machine learning, and optimization techniques in networks. In particular, he is interested in the broad areas of wireless networks, cognitive networking, software-defined networks, and cloud computing. He serves as an Associate Editor for IEEE Access, IEEE Communication Magazine, and Springer Nature Big Data Analytics. He is a member of ACM, and a senior member of IEEE.

Adnan Khalid Kiani is currently working as Assistant Professor at National University of Sciences and Technology Pakistan. He completed his Ph.D. and Master's degrees in Network Communications back in 2005 and 2009 respectively from Brunel University, London. Prior to that he did his Undergraduate degree in Electrical Engineering from National University of Sciences and Technology, Pakistan. His research interests include network communications, mobile ad hoc and wireless sensor networks, routing issues in networks, and network security.

Raihan ur Rasool has been working at King Faisal University, Saudi Arabia as Assistance Professor. He has recently joined Victoria University Melbourne and is on long leave from National University of Sciences and Technology (NUST) Pakistan. He earned Ph.D. in Computer Engineering from the Wuhan University of Technology China, under Cultural Exchange Scholarship and completed his post-doctorate at the University of Chicago, USA on prestigious Fulbright fellowship by the Department of State. He has numerous honors and awards to his credit, and best Ph.D. student award is one of them. His research interests include large-scale systems, security and computer architecture. His research work, comprising over 40 articles is published in various international conferences and journals. He has published in leading avenues of research like ISCA, HiPEC, CCGrid, ACM SIGARCH, IEEE Transactions on Cloud Computing, JNCA & FGCS.



Usman Younis received his B.E. and M.S. degrees from the National University of Sciences and Technology, Islamabad, Pakistan, in 2004 and 2006, respectively, and the Ph.D. degree in electronics and electrical engineering from the University of Glasgow, Glasgow, U.K., in 2010. He was with the Maritime Technologies Complex, Islamabad, as an Assistant Manager (Technical) from 2004 to 2006, and with the Department of Electrical and Computer Engineering, National University of Singapore, as a Research Fellow from 2015 to 2016. He holds the position of Assistant Professor at the National University of Sciences and Technology since 2010.