Network and Complex Systems
ISSN 2224-610X (Paper) ISSN 2225-0603 (Online)
Vol.12, 2021

Intrusion Detection System in Cloud Computing: A Literature Review

Kezang Dema[1*] Naghmeh Moradpoor[2] Thinley Jamtsho[3]

1. Information Technology Department, College of Science and Technology, Royal University of Bhutan, Chhukha, Bhutan
2. School of Science, Engineering and Technology, University of Abertay , Dundee, UK
3. Department of Science and Mathematics, Phuentsholing Middle Secondary School, Phuentsholing, Chhukha, Bhutan

* E-mail of the corresponding author: kelden.dema@gmail.com

Abstract

Cloud computing provide users a convenient, ubiquitous and on-demand network access to a shared pool of computing resources over an Internet. On the other aspect, due to its distributed and complex architectures, it possess many security risks and become one of the attractive targets for the cyber-attacks by the intruders. With the growing popularity of cloud computing, the importance to provide reliable and secure services assurance remains one of the major issues. Intrusion Detection Systems (IDS) have been widely used as one of the mechanism to detect attacks on cloud. Different authors have proposed various IDS that can be used in cloud computing in an attempt to provide secure services.

This paper surveys different methods and approaches applied in intrusions detection systems (IDS) of cloud environment by various authors. The paper is organized as follows. Section 1 discusses about the different kinds of attacks or intrusions found in cloud. Section 2 discusses and demonstrates different IDS used in cloud that have been proposed by various authors in their research. Section 3 concludes the review with references at the end.

Keywords: Cloud computing, IDS, NIDS, HIDS, DIDS
DOI: 10.7176/NCS/12-05
Publication date: January 31st 2021

1. Introduction

Cloud computing is one of the growing technology that provides a framework for end users in enabling a powerful cloud services and applications on-demand through Internet (Dhage *et al*. 2011). The cloud provides services such as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). These cloud services are provided through internet, it may easily be exposed to the risk of security attacks. Therefore, security has become the greatest challenge of cloud computing. In cloud computing, security issues such as confidentiality, integrity and availability (CIA) are the most important security consideration (Lo *et al*. 2010).The major security concern after data security is the network security. It needs to look into the security of networks from various attacks like IP spoofing, Denial-of-service (DOS) and Distributed Denial-Of-Service (DDOS) (Alqahtani *et al*. 2014). The cloud should incorporate efficient Intrusion Detection System (IDS) to resist these kinds of attacks. An Intrusion Detection System refers to a hardware device or a software application that monitors network or system activities for malicious intrusions (Dhage *et al*. 2011).

The main objective is to conduct a survey on various IDS of cloud computing. Cloud environment is targeted by the attackers with various intrusions which will be explained in the following section. The need of security in the cloud has led to the emergence of different intrusion detection system. The survey presents different authors who proposed various IDS based on different algorithms, approaches, techniques and others to help protect the cloud environment. In this section, some of the attacks affecting the confidentiality, integrity and availability of cloud services will be briefly discussed.

1.1. DoS and DDoS Attack

DoS and DDoS are common security attacks where the attackers' main motive lies in making the resources of the victim devices unavailable to its authorized users. In this type of attack, a bunch of data packets i.e SYN flooding, UDP flooding, and ICMP flooding are launched against the victim's devices continually (Lo *et al*. 2010).

1.2. Port Scanning

Port scanning methods gives a detail of open ports and closed ports. An attacker may perform port scanning and attack on services running on those open ports. Through this type of attack, critical information related to network such as IP address, MAC address, rules, etc can be exposed.

1.3. Attacks on VM and Hypervisor

In cloud scenario, we have many virtual machines which run on hypervisors. Some of the common attacks on virtual layers are SubVir, DKSM, BLUEPILL and many others. In this type of attack, the installed hypervisor is compromised and the attacker gains control over the host.

2. Literature Review

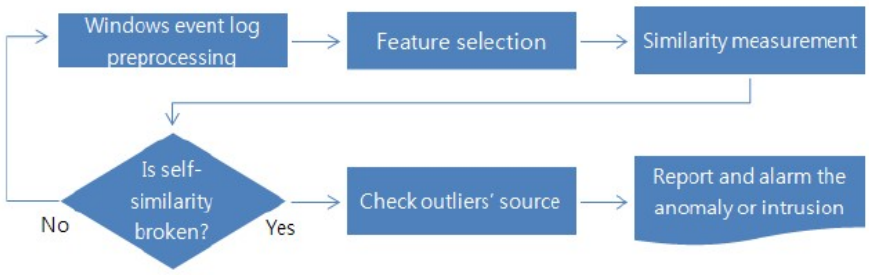
The cloud computing use mainly four different types of IDS namely: Host based intrusion detection system (HIDS), Network based intrusion detection system (NIDS), Hypervisor based intrusion Detection System and Distributed intrusion detection system (DIDS).

2.1. Host Intrusion Detection System (HIDS)

The host-based intrusion detection system collects the data from the specific host machine (Kwon *et al*. 2011). The data includes such as file system used, system calls, network events, system log files, etc which are monitored and analyzed by HIDS for intrusions. In cloud, HIDS are place in different places: on a host machine, hypervisor or virtual machines (VMs). The intrusive behaviors are detected through monitoring and analyzing log file, security access control policies and user login information (Modi *et al*. 2013). HIDS will be monitored by cloud provider if it is installed on hypervisor and by the cloud user if it is installed on VM (Cox 2011).

The *self-similarity based lightweight intrusion detection method for cloud computing* (Kwon *et al*. 2011) makes use of self-similarity observed in system's internal activities such as system calls and process status. When there isn't intrusion, every internal events will generate some similarity patterns since it depend on the usage pattern of users, processes and various applications (Kwon *et al*. 2011). But, when intrusions occur, there will be deviation in the normal patterns. In this proposal, the system's self-similarity is monitored to detect the system's anomalies. The procedures of the proposal are depicted in Figure 1.The windows event log preprocessor extracts the number of events from the windows' security event. The feature selection procedure makes group by combining security ID (SID) and EventID (EID) in windows system.

The self-similarity uses two techniques namely cosine and hybrid for the calculation. The self-similarity for each VM is measured and if the calculated similarity observes some deviation from normal behavior, IDS generates alerts and informs the system administrator. The outlier source identifies the source IP address and the intruder who made the abnormal event. The detected intrusion information is reported to the system administrator.



Intelligent Information and Database System (Kwon *et al*. 2011, p 358).

Figure 1. Procedures of self-similarity based lightweight IDS.

This approach results to be cost-effective because the IDS don't need a long learning process that would require many system resources and high cost. And, it is one of the most efficient approach in detecting anomaly with no complex algorithm requirement in cloud computing. But the approach works only for windows system and it is not feasible for other systems.

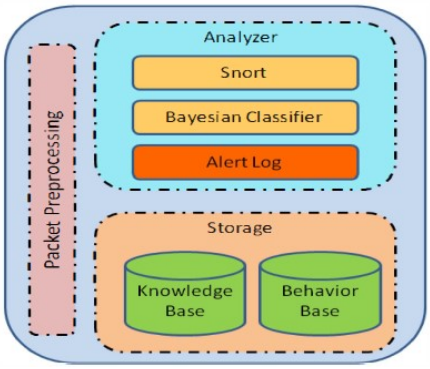
Alqahtani *et al*. (2014) came with an *intelligent intrusion detection system for cloud computing* which helps in detecting DoS and DDoS attacks in a private cloud named SaaSCloud. The intelligent SaaS intrusion detection system in Cloud Computing (SIDSCC) system was modeled by conducting a survey regarding cloud computing and IDS from six different countries. The surveyed closed-ended questionnaires were later used for quantitative analysis. The model was implemented in three virtual machines namely SaaSCloud Model, DDoS attack Model and IDSServer Model. The SaaSCloud was created by deploying a php webpage with Apache used as a web server. This page was used by attacker to implement and send the ICMP flooding on it. Snort works as an IDS Server that will monitor the traffics and detect malicious activities on SaaSCloud. In the event of ICMP flood attack on SaaSCloud, it requests the IDSServer to detect and ping the IP address of attacker to block the activity. The administrator sends an alert of the attack to the SaaSCloud. The IDSServer analyzes the console for intrusion databases and sends the malicious IP address to cloud service provider to act upon it.

The result of this proposed model was positive as it was able to detect attacks especially the ICMP floods and alarms the user and IDS Server admin. But, like most of the models, it doesn't detect the unknown attacks and it only guarantees the detection of known attack especially the ICMP flooding. And also, the system fails to detect the intrusion whereby legitimate users cannot access SaaSCloud services if the ICMP attacking rate is higher than 6000 pps. And model would work only for SaaSCloud.

2.2. Network Intrusion Detection System (NIDS)

Network Intrusion Detection System (NIDS) identifies intrusions or malicious activities such as DoS attacks, DDOS attacks, port scans and others by examining network traffic flowing among multiple hosts connected to the network. NIDS can be hardware sensors that are usually located at various points along the network or software that is installed on various computers connected along the network. The intrusions are detected using both signature and anomaly based techniques. Unlike HIDS, it has limited visibility inside the host machines.

Modi *et al*. (2012) proposed a *Bayesian Classifier and Snort based NIDS in cloud* to detect network attacks. The model used Bayesian classification algorithm to predict the unknown attacks by observing previously stored network events and snort is used to detect known attacks. The proposed NIDS consists of three components viz; packet preprocessing, analyzer and storage as shown in Figure 2.



Bayesian classifier and snort based network intrusion detection system in cloud computing (Modi *et al*. 2012, p3).

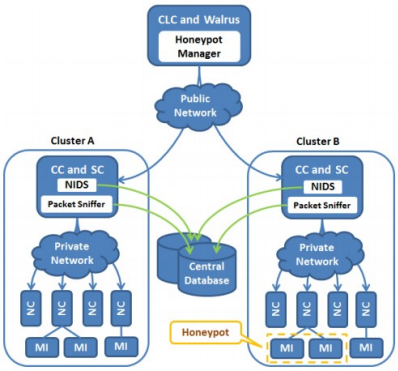
Figure 2. Architecture of Bayesian classifier and snort based NIDS.

The packet preprocessing component processes captured packets and removes redundant information that has low correlation with detection (Modi *et al*. 2012). Analyzer component consist of snort, Bayesian classifier and Alert Log. The main function of the analyzer is to apply the detection techniques on captured packets. Snort uses the signature based detection technique and detects known intrusions. The alert log component logs the intrusion event and sends alert message to the storage component for storage. Next, the non-intrusion packets are preprocessed for anomaly detection by observing behavior base using Bayesian classifier.

The intrusion detected is logged into alert log database and only the normal packets are allowed into the system. The storages component has knowledge base which stores the rules related to known attacks used by snort and the behavior base which stores network events that are used by Bayesian classifier. The proposed model is able to detect higher number of both known and unknown intrusions. It provides low false positives and false negatives. The model only detects the intrusion coming from network traffics but this approach cannot detect any

intrusions which are running on VMs.

Borisaniya *et al*. (2012) introduced *honeypot in cloud IDS* design to help in detecting potential attacks with reduced number of false positives. A honeypot is a deception system which allures the attackers (Borisaniya *et al*. 2012). The open source cloud computing framework known as Eucalyptus is being implemented where the incorporation of honeypot with IDS is easy. A eucalyptus has four components: Cloud Controller (CLC), Node Controller (NC) which manages, Cluster Controller (CC) and Storage Controller (SC). The NIDS are placed in cluster controller to monitor the network traffic. The packet sniffer sniffs and logs all network packets whose source or destination IP is one of the honeypot instances in the central database. The honeypot instances are placed through the honeypot manager which is placed in cloud controller (CLC) as shown in Figure 3. The main idea of introducing honeypot is to monitor those network packets which was previously marked as legitimate packets by NIDS, and then to identify suspicious activities on that same packets.



Trust management VI (Borisaniya *et al*. 2012, p 89).

Figure 3. Architecture of intrusion detection system using honeypot.

Snort is used along with packet sniffer in CC listens to all the traffics including that traffic intended towards honey pot instances, and it generates alerts to the central database if any intrusion occurs. The honeypot is created using a set of machine images of different OS that can attract the attackers. Those images are monitored periodically for intrusion attempts and if the intrusion occurs, an alert is reported.

The proposed system detects known and potential unknown attacks. And the controlled used of honeypot adds in generating less number of false alarms. The system seems to detect more of known attacks compared to unknown attacks. And studying the architecture, it seems it is not cost effective and will incur more system resources as there is the requirement of honeypot instances.

Modi *et al*. (2012) proposed a framework of *integrating signature apriori algorithm and snort based network intrusion detection system*. The NIDS are positioned in different location either in front end or in back end of the cloud. The NIDS consists of snort and signature apriori algorithm. The snort captures packets and implements a signature based technique to detect network intrusions. The signature apriori algorithm takes in captured packets and partially known signatures and generates new attack signatures. The snorts capture network packets and make decisions, either to allow or deny based on its predefined configured rules. And also, captured packets along with partially known attack signatures are given to the apriori algorithm. The algorithm generates new possible signatures and these are appended in the snort configuration file as new rules that help snort in detecting known attacks as well as derivative attacks.

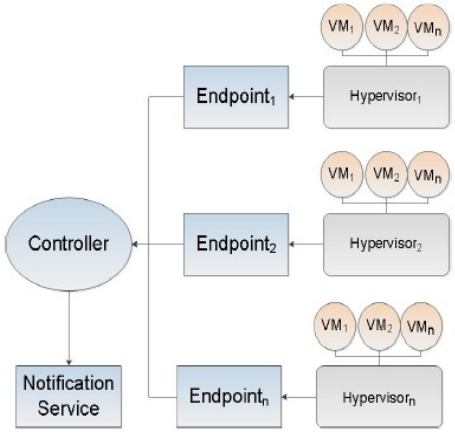
The proposed framework detects known attacks and the apriori algorithm helps in detecting derivative of known attacks by updating new rules in snort. It was also found out that the framework has lower false alarm rate. And the framework had low cost of computation because the signature apriori generates rules once and it does not require more number of NIDS instances. The proposed model makes use of snorts that will definitely help in detecting known attacks but there seems low probability of detecting the new attacks.

2.3. Hypervisor based Intrusion Detection System

The hypervisor-based intrusion detection system is a kind of IDS which runs on a hypervisor layer platform. It

allows user to monitor and analyze communications between virtual machines, between hypervisor and VM and within the hypervisor based virtual network. The one of the benefits of this type of IDS is the availability of information.

Nikolai & Wang (2014) proposed *architecture to perform intrusion detection security using performance metrics obtained from hypervisor*. The performance metrics like network data transmitted, network data retrieved, CPU utilization and others used in the architecture are directly gathered from the hypervisors hosting virtual machines within the cloud environment. The proposed framework has three main components: controller node, end point nodes and a notification service as shown in Figure 4.



Hypervisor-based cloud intrusion detection system (Nikolai & Wang 2014, p 990).

Figure 4. Framework of Hypervisor based IDS.

The controller endpoint nodes gather the performance metrics data from hypervisors and format them to send it to the controller node. The controller node would collect the data from endpoint nodes and analyze using a sliding window approach. The notification service components provide alerts if the system detects a signature of a potential attacks. The experiment was conducted on the Eucalyptus infrastructure and it showed that the DoS attacks could be detected by streaming hypervisor performance metrics.

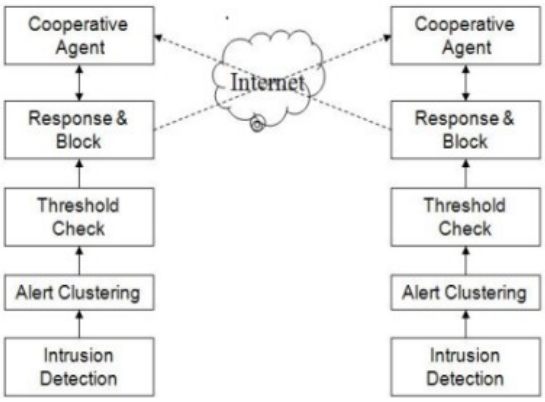
This proposed system does not require other additional software installed in virtual machines and moreover, the monitoring is done outside the VM and is independent of the operating system and other applications within the virtual machines. The significant advantage of the system is the detection of insider attacks but, the system is not able to handle the attacks from outside.

2.4. Distributed Intrusion Detection System (DIDS)

A Distributed Intrusion detection System (DIDS) consists of both HIDS and NIDS over a large network, all of which communicate with each other, or with a central server that enables network monitoring (Modi 2013). DIDS discovers attacks on individual hosts as well as the network which connects them. This type of IDS uses a combination of anomaly and signature based detection technique for the detection and analysis of intrusions. In cloud environment, it can be placed at host machine or at the processing server in backend.

The *cooperative based IDS for cloud computing environment* is to reduce the impact of denial-of-service (DoS) attack or distributed denial-of-service (DDoS) attack (Lo *et al*. 2010). The proposed system deploys IDS in each cloud computing region and this particular proposal aims at supporting an idea of cooperative defense by all the IDS. If any of the cloud regions detects intrusion, the IDS in it alert other region about the attack. And the severity of the alert is being judged based on certain criteria. If the intrusion detected is regarded as new kind of attacks, then the new blocking rule is added into the block table. Hence, this type of early detection and cooperation among IDS help in resisting attacks in cloud.

This architecture has five components within IDS system: intrusion detection, alert clustering, threshold check, response and block and cooperative agent as shown in Figure 5.



A cooperative intrusion detection system framework for cloud computing networks (Lo *et al*. 2010, p 281).

Figure 5. Cooperative based IDS.

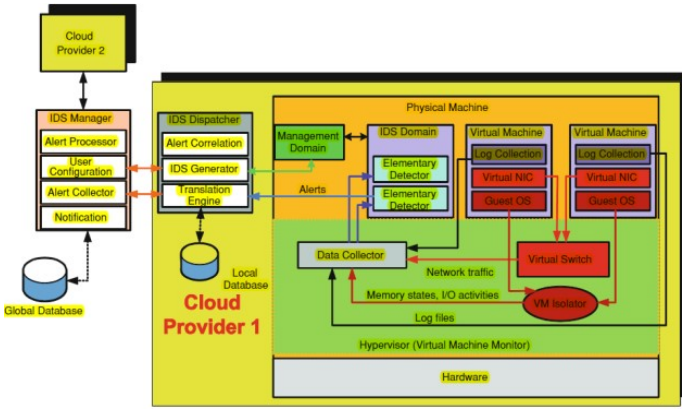
The intrusion detection collects and analyzes network packets. If the packets match with the one listed in the block table, it is dropped and sends alert message about the detected attacks to other regions. Alert clustering collects alert produced by other regions and identify the level of spacious packed based on three alert levels viz: serious, moderate, and slight. The threshold check is performed on alert message about the packet. It uses data clustering method to find outlier. The decision about alert on whether it's true of false is identified after calculating the severity. The response and blocking component blocks bad packets and sends alert notification to other IDSs.

The proposed cooperative IDS prevent the cloud system from single point of failure attack caused by DoS and DDoS attacks. The intrusion detection is purely based on the signature based technique where only the known attacks could be detected. It probability of detecting unknown attack is very low.

Ram (2012) *proposed mutual agent based approach to detect intrusion* particularly DDoS attacks in cloud. The IDS module is placed in each cloud region. Snort is used for intrusion detection. And also, each region has mutual agents that have the responsibility to notify other regions if there is any intrusion taking place. The alerts generated from other regions would be calculated by all regions to check the severity. After the calculation, if new attack or intrusion is found, new signature rule would be added into the block table at each region. There exists a mutual cooperation among cloud regions that enables to detect attacks especially the DDoS attack in whole cloud region.

The proposed approach could detect the DDoS attacks that would help cloud system from single point of failure attack (Lo *et al*. 2010). The approach seems easy to implement and not complex, but in reality, network is always targeted with both known and unknown attacks. In this approach, only Snort is deployed which would only detect the known attacks by finding similar signatures in block table. Hence, this approach is not able to detect unknown attacks and moreover, it requires high computation cost for exchanging alerts.

Man & Huh (2012) proposed *collaborative IDS for cloud computing*, a system that would help in reducing the impact of large-scale coordinated attacks such as DDoS, worms and stealthy scan by notifying any new intrusions to cloud users' systems. The proposed IDS model have three main parts namely, elementary detector, IDS Dispatcher and IDS Manager.



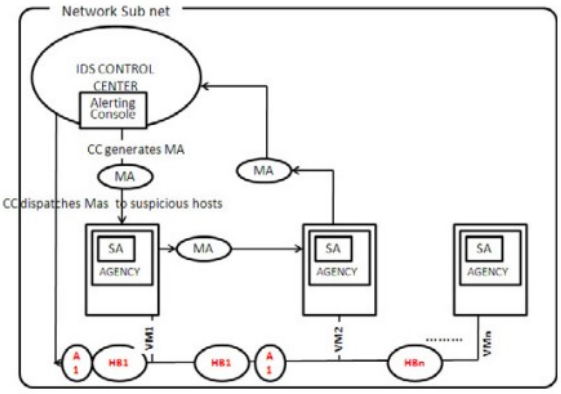
A collaborative intrusion detection system framework for cloud computing (Man and Huh 2012, p 99).
Figure 6. Architecture of CIDS.

The elementary detectors are specialized IDS that resides in each VM to monitor the machines and generate alarms when intrusion is detected. The IDS dispatcher is placed inside each cloud region and the IDS manager is located at the management region of a collaborative cloud. Along with those components, it has global a local database servers. The global database server resides in IDS manager region and the local database servers in each IDS dispatcher. The elementary detectors (EDs) residing in distributed VM collects and analyzes data about the network traffic, memory files, logs and others. In the detection of intrusions using unsupervised anomaly detection technique, it generates raw alerts and sent to IDS Dispatcher for alert aggregation and correlation. The IDS Dispatcher would then, process raw alerts collected from all EDs. It would aggregate and correlate all raw alerts from EDS into hyper alerts and analyze them to detect large-scale coordinated attacks.

This component is also responsible to convert those alerts into common format and store in the local database. The alerts are aggregated to create hyper alerts and it gets forwarded to IDS manager. IDS manager gathers all information or events related to intrusions and send notifications to users through one single interface for all cloud regions.

The proposed system improved the efficiency of intrusion detection over a large-scale environment. Moreover, the ID resources are shared between networks, which potentially reduce computational cost. The number of false alarms generated by individual IDS is reduced. The aggregation of alarms generated by different IDSs to create hyper alerts produced more comprehensive information about intrusion attempts compared to that of attained using single IDS. The proposed system allows various cloud service providers to work together for the detection of intrusions but this may add heady loads on a central intrusion detection server.

Dastjerdi *et al*. (2009) proposed *the use of mobile agents in detecting attacks or intrusions for cloud applications* regardless of their locations. The proposed hybrid model places IDS in each subnet of virtual machines as show in Figure 7.



Distributed intrusion detection in clouds using mobile agents (Dastjerdi *et al*. 2009, p 178).

Figure 7. A mobile agent based IDS.

The model consists of four main components namely, IDS Controller Center (IDS CC), Agency, Specialized Investigative Mobile Agent and Application Specific Static Agent Detectors. In brief, IDS CC is the central point of IDS components which include other components like databases of all intrusion patterns, alerting console, agent generator, mobile agent generator and many others. The static agents (SA) generate an alert whenever any suspicious activities are detected. The main function of the Mobile agent (MA) is to visit and investigate all the VMs that sent similar alerts of suspicious activities detection. It collects information, correlates it and then sends the result to IDS Control Center.

When there are any suspicious activities, Static Agents (SA) generates an alert, save that information in a log file and send alert's ID to IDS control center. Then, the control center send task-specific mobile agent to every agency that sent similar alerts. As depicted in the above figure 8, MA will investigate all those VMs, get information and correlated that information and then, carry back the result to IDS CC.

The result will be analyzed and compared with intrusion patterns in IDS CC database by the alerting console. Alarm will be raised if intrusion is detected and IDS CC saves the information received from Mobile agent (MA) into its database. The names and identification of discovered VM will be black listed and sent to all other VMS except the black listed VMs. The model provides the benefits of scalability, flexibility and cost-effectiveness. The system can be operated in heterogeneous environments with lower operational cost (Man & Huh 2012). The model protects and detects intrusion in VMs outside the organization but it produces more network load which is not desired.

3. Conclusion

The cloud computing provides end users a powerful cloud services and applications on-demand through Internet. The different cloud services that the users get are software-based, platform-based and infrastructure-based. The availability and abundance of cloud resources over the Internet leads in welcoming different attacks. The security becomes the major issues in cloud computing that require considerable attention. The IDS implemented based on different methods and approaches proposed by various authors' acts as one of the measure to handle the security issues in cloud.

References

Alqahtani, S.M., Balushi, M.A., & John, R. (2014). An intelligent intrusion detection system for cloud computing (SIDSCC). *International Conference on Computational Science and Computational Intelligence IEEE,* 135-141

Borisaniya, B., Patel, A., Patel, D.R., & Patel, H. (2012). Incorporating honeypot for intrusion detection in cloud infrastructure. *IFIP Internaltional Conference on Trust management*, 84-96.

Dastjerdi, A.V., Bakar,K.A., & Tabatabaei, S.G.H. (2009). Distributed intrusion detection in clouds using mobile agents. *Advanced Engineering Computing and Applications in Sciences,* 175-180

Dhage, S.N., Meshram, B.B., Rawat, R., Padawe, S., Paingaokar, M., & Misra, A. (2011). Intrusion detection system in cloud computing environment. *ICWET'11 Proceedings of the International Conference &*

Workshop on Emerging Trends in Technology, 235-239.

Kwon, H., Kim, T., Yu, S.J., & Kim, H.K. (2011). Self-similarity based light weight intrusion detection method for cloud computing. *Asian Conference on Intelligent Information and Database Systems*, 353-362.

Lo, C., Huang, C., & Ku, J. (2010). A cooperative intrusion detection system framework for cloud computing networks. *IEEE International Conference on Parallel Processing Workshops (ICPPW),* 280-284.

Man, N. D., & Huh, E. (2012). A collaborative intrusion detection system framework for cloud computing. *Proceedings of the International Conference on IT Convergence and Security 2011*, 91-109.

Modi, C., Patel, D., Borsaniya, B., Patel, H., Patel, A., & Rajarajan, M. (2013). A survey of intrusion detection techniques in cloud. *Journal of Network and Computer Applications,* 36, 42-57.

Modi, C.N., Patel, D.R., Patel, A., & Muttukrishnan, R. (2012). Bayesian classifier and snort based network intrusion detection system in cloud computing. *2012 Third International Conference on Computing Communication & Networking Technologies*, 1-7.

Modi, C.N., Patel, D.R., Patel, A., & Rajarajan, M. (2012). Integrating signature apriori based network intrusion detection system (NIDS) in cloud computing. *Procedia Technology,* 6, 905-912.

Nikolai, J., & Wang, Y. (2014). Hypervisor-based cloud intrusion detection system. *IEEE International Conference on Networking and Communications (ICNC),* 989-993.

Ram, S. (2012). Secure cloud computing based on mutual intrusion detection system. *International Journal of Computer Application, 2,* 57–67.