Copyright © 2018 Tech Science Press CMC, vol.57, no.2, pp.297-306, 2018

Multi-VMs Intrusion Detection for Cloud Security Using Dempster-shafer Theory

Chak Fong Cheang[1, *], Yiqin Wang[1], Zhiping Cai[2] and Gen Xu[1]

Abstract: Cloud computing provides easy and on-demand access to computing resources in a configurable pool. The flexibility of the cloud environment attracts more and more network services to be deployed on the cloud using groups of virtual machines (VMs), instead of being restricted on a single physical server. When more and more network services are deployed on the cloud, the detection of the intrusion likes Distributed Denial-of-Service (DDoS) attack becomes much more challenging than that on the traditional servers because even a single network service now is possibly provided by groups of VMs across the cloud system. In this paper, we propose a cloud-based intrusion detection system (IDS) which inspects the features of data flow between neighboring VMs, analyzes the probability of being attacked on each pair of VMs and then regards it as independent evidence using Dempster-Shafer theory, and eventually combines the evidence among all pairs of VMs using the method of evidence fusion. Unlike the traditional IDS that focus on analyzing the entire network service externally, our proposed algorithm makes full use of the internal interactions between VMs, and the experiment proved that it can provide more accurate results than the traditional algorithm.

Keywords: Intrusion detection, cloud computing, Dempster-Shafer theory, evidence fusion.

1 Introduction

Cloud security is one of the most important factors in cloud computing [Chonka, Xiang, Zhou et al. (2011); Khorshed, Shawkat and Saleh (2012); Guo, Liu, Cai et al. (2018)]. The widely used of virtualization on the cloud environment make the network situation becomes more complicated than before. Many network services now deployed on the cloud are distributed across different virtual machines (VMs). This makes it easier in suffering from distributed attacks and more different in detecting the intrusions.

Intrusion detection system (IDS), used to protect traditional network systems, now can be deployed on the cloud system as one of the most effective methods of protection. In recent years, many improvements on the algorithms of IDS are proposed. Modi et al. [Modi, Patel, Borisaniya et al. (2013)] proposed a collaborative IDS framework for cloud. In this proposed collaborative IDS, cascading decision tree and SVM are used to improve

[1] Faculty of Information Technology, Macau University of Science and Technology, Macau.

[2] College of Computer, National University of Defense Technology, Changsha, 410073, China.

[*] Corresponding Author: Chak Fong Cheang. Email: cfcheang@must.edu.mo.

CMC. doi:10.32604/cmc.2018.03808 www.techscience.com/cmc

the detection accuracy and the system performance. Vaid et al. [Vaid and Verma (2015)] proposed an IDS using Bootstrapped Optimistic Algorithm for Tree Construction (BOAT) algorithm. This research project is aimed to analyze the user behavior using anomaly detection of malicious activities when unauthorized access or illegal transactions to cloud data occurred. Ficco et al. [Ficco, Tasquier and Aversa (2013)] proposed a distributed intrusion detection architecture, which allows the cloud providers to offer the security solutions as a service. Mishra et al. [Mishra, Pilli, Varadharajan et al. (2017)] proposed a combination of parallelization and machine learning methods, which enhances both the detection mechanism and the detection speed of an IDS. Khan et al. [Khan, Awad and Thuraisingham (2007); Mewada, Gedam, Khan et al. (2010); Chang, Li and Yang (2017)] proposed the improved SVM algorithms for intrusion detection classification. Mukkamala et al. [Mukkamala, Janoski and Sung (2002); Abhaya and Kumar (2016); Bezdek and James (1981)] tried to solve the intrusion detection problem using neural network and fuzzy algorithms. Gul et al. [Gul and Hussain (2011); Singh, Patel, Borisaniya et al. (2016)] proposed some distributed models and collaborative frameworks for intrusion detection in the cloud.

However, there are still many limitations when merely migrating the traditional IDS onto the cloud. For network-based IDS (NIDS), it gives better observation and more resistibility against offending attacks but lacks the knowledge about host system. On the other hand, host-based IDS (HIDS) provides security against the host system but still cannot detect and resist attacks on other hosts or network, and are vulnerable to evasion attacks. More important, the traditional IDS usually focuses on analyzing the entire network service externally but neglects the internal interaction between the VMs.

Therefore, we propose a cloud-based IDS using the Dempster-Shafer theory to overcome the limitation of the traditional IDS. In this system, each VM can observe the malicious activities and independently analyze them using its own IDS algorithm. Then the cloud-based IDS can combine the results of all VMs using the method of evidence fusion. This IDS system makes full use of the internal relationship of the VMs in analyzing the intrusion detection, therefore it can improve the accuracy of the judgment.

The paper is organized as follows. In Section 2, we formulate the problem in intrusion detection. In Section 3, we describe the IDS architecture and its components. In Section 4, we explain the IDS algorithms using the Dempster-Shafer theory. In Section 5, we discuss the simulation experiment of the multi-VMs IDS. Finally, we make a conclusion about our work in Section 6.

2 Problem formulated

In cloud computing systems, intrusion detection algorithms are used to recognize intrusion activities by monitoring the network traffic and the abnormal events, and the measurement of intrusion activities is regarded as evidence. Distributed intrusion detection system can obtain the evidence from individual observer and provide a numerical procedure for combining multiple pieces of evidence from different hosts. Thus the intrusion detection is essentially a kind of pattern classification problems.

There are many methods for combining evidence, such as simple majority voting, simple majority decision rule, averaging the observers' numerical evidence and etc. Among most of the proposed evidence fusion methods, Bayesian approach interprets the posteriori probability $P(H|E)$ as a measure of belief about a hypothesis H updated in response to evidence E. Bayesian approach is well grounded in the formalities of probability through the well-known Bayes' theorem.

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} \tag{1}$$

However, one difficulty in Bayesian approach is the requirement to know the priori probability in the absence of any evidence, because it requires complete knowledge of both prior and conditional probabilities.

Dempster-Shafer theory is considered to be an extended Bayesian inference. The Dempster-Shafer theory of evidence, originated by Dempster [Dempster (1976)] and later revised by Shafer [Shafer (1976)] addresses this situation by representing uncertainty in the form of belief functions. It offers a mathematical way to combine evidence from multiple observers without the need to know about a priori or conditional probabilities as in the Bayesian approach. It has solved the problem of analyzing the uncertainty in a quantitative way by representing them using belief functions. Therefore, when it is used in the distributed intrusion detection, Dempster-Shafer theory can produce the results as malicious intrusions or normal activities with an unknown bias.

3 Architecture of multi-VMs IDS

When deploying the network services on multiple VMs across the cloud system, it is vulnerable to a variety of malicious attacks and is difficult to detect it. For this purpose, we design an architecture of a cloud-based IDS, in which individual VM can observe part of the total traffic, and make distributed intrusion detection.

3.1 Network services in IDS

This section will explain the components of proposed IDS architecture, and provide the definition of each component. As shown in Fig. 1, our proposed IDS is expected to use on a cloud system that consists of multiple physical servers. Each physical server is indicated as:

$$PS_i, i = 1,2, \dots, n \tag{2}$$

In each physical server, there are some virtual machines (VMs) hosted inside it. Each VM is indicated as:

$$VM_{i,j}, i = 1,2, \dots, n; j = 1,2, \dots, m \tag{3}$$

Where i is the index of the physical server, and j is the index of VM in this PS_i.

We supposed that a specific network service deployed on the cloud can be distributed across different VMs. The provision of a specific network function is actually provided by the data flow throughout different VMs. For example, a given network service can be a typical network function designed for database enquiry and is implemented in server/client mode. This network service might consist of multiple servers, such as a web server, an authentication server, a database server, and some other servers. Each network service is indicated as:

$$NS = \{VM_{i1,j1} \rightarrow VM_{i2,j2} \rightarrow \cdots \rightarrow VM_{ik,jk}\} \tag{4}$$

$VM_{ik,jk}$ can be simplified as VM_k, where k is the index of VMs hosted this network service.

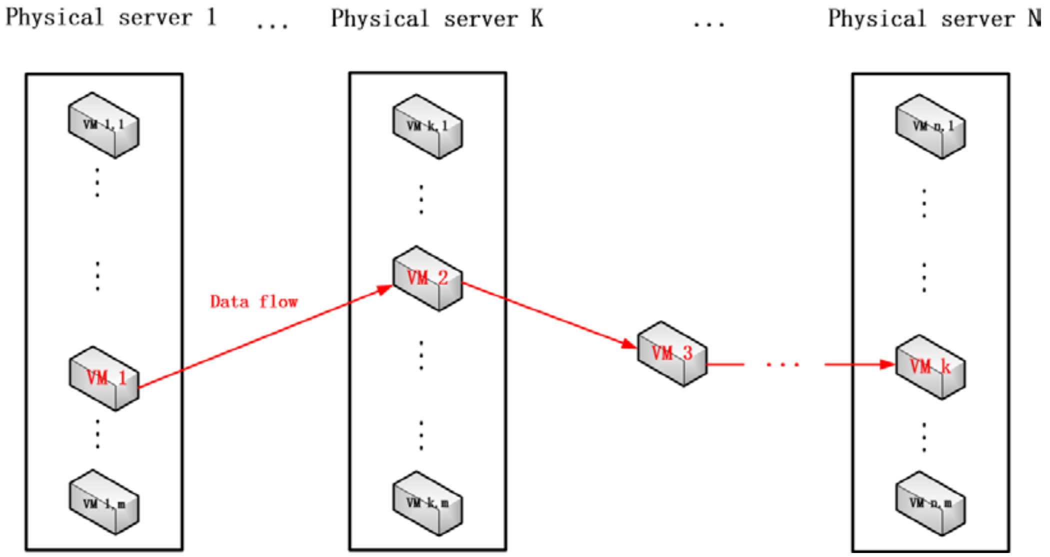

Figure 1: A network service is deployed on multiple VMs across physical servers

3.2 Evidence in IDS

One of major problem in distributed intrusion detection is to define the trustworthiness of the hosts and to combine the observational evidence from multiple hosts. In our proposed IDS, evidence is the likelihood of being attacked on a VM, which is generated by the existing intrusion detection algorithm, such as k-nearest neighbor (KNN), support vector machine (SVM), decision tree (DT) and etc. The evidence of each VM is indicated as:

$$E(VM_{i,j}), i = 1,2, \dots, n; j = 1,2, \dots, m \tag{5}$$

Therefore, the evidence of a specific network service is the likelihood of being attacked on either of VMs, and it is the fusion of evidence obtained by multiple VMs. The evidence of a specific network service is indicated as:

$$E(NS) = Fuse\left(E(VM_{i1,j1}), E(VM_{i2,j2}), \dots, E(VM_{ik,jk})\right) \tag{6}$$

For a specific network service, the data flow is the traffic of messages that created by the end users and is continuously forwarded throughout all VMs involved in this network service. More precisely, each message is forwarded from the first VM hop by hop until it reaches the last VM. During the forwarding process, the message might be changed from VM to VM, in order to provide necessary function to the end users.

Considering the fact that any message is possibly changed at any VM during the forwarding, but the message always remains the same after it is sent by the previous VM and before it is received by the next VM. The evidence is generated on these two different VMs based on the same snapshot of the observed message. Therefore, two neighboring

VMs can be grouped as a pair of observers for evidence fusion first, and it is reasonable to break down the procedure of evidence fusion into two levels, as shown in Fig. 2.

The evidence fusion is designed in two levels.

At the first level, the VM-based evidence is fused. The evidence of each pair of VMs is generated by combining the evidence of two neighboring VMs, and is indicated as:

$$E\left(NS_{k-1,k}\right) = Fuse\left(E\left(VM_{ik-1,jk-1}\right), E\left(VM_{ik,jk}\right)\right) \tag{7}$$

At the second level, the Cloud-based evidence is fused. The evidence of all VMs is generated by combining the evidence of all pair of VMs, and is indicated as:

$$E(NS) = Fuse\left(E\left(NS_{1,2}\right), E\left(NS_{2,3}\right), \dots, E\left(NS_{k-2,k-1}\right), E\left(NS_{k-1,k}\right)\right) \tag{8}$$

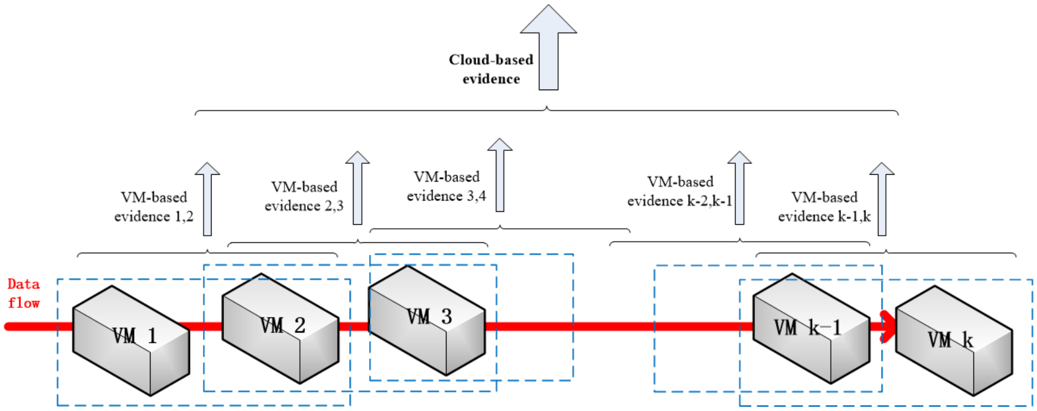

Figure 2: Evidence fusion in multi-VM IDS

4 Algorithms of Multi-VMs IDS using Dempster-Shafer theory

Detection accuracy is an issue for any intrusion detection system. When estimating the likelihood of an intrusion from multiple hosts, the decision of the individual host might not be reliable. The Dempster-Shafer theory of evidence is well suited for this type of problem because it reflects uncertainty.

4.1 Evidence definition

The definition of evidence is the problem of determining initial estimates of hosts' trustworthiness. In our proposed IDS, the evidence can be generated by using any existing intrusion detection algorithms. Sometimes, we can compute an initial estimate of the hosts' trustworthiness by combining multiple classifiers such as k-nearest neighbor (KNN), support vector machine (SVM), random forest, decision tree (DT) and supervised learning in quest (SLIQ), because these techniques have low false alerts, better accuracy and low computation cost.

First, we define Ω as all possible types of malicious attacks on the cloud. Here, Ω is a collection of mutually exclusive and finite elements. Each element in the set represents one type of malicious attack. Various types of DDoS attacks, e.g. TCP SYN flood, UDP flood, ICMP flood and etc. are indicated as:

$$\Omega = \{a_1, a_2, \dots, a_n\} \tag{9}$$

The set of all subsets of Ω is called power set of the malicious attacks, denoted as 2^{Ω}.

$$A \equiv 2^{\Omega} = \{A_1,\ A_2, \dots, A_m\} \tag{10}$$

$$A_i \in \{\emptyset, \{a_1\}, \{a_2\}, \dots, \{a_1, a_2\}, \{a_1, a_3\}, \{a_2, a_3\}, \dots, \Omega\} \tag{11}$$

Where A_i is a member of the power set, and it is the subset of all malicious attacks.
Second, the mass function $m()$ is defined to measure the likelihood of any malicious attacks.

$$m: 2^{\Omega} \rightarrow [0,1] \tag{12}$$

Meet:

$$m(\emptyset) = 0 \tag{13}$$

$$\sum_{A_i \subseteq 2^{\Omega}} m(A_i) = 1 \tag{14}$$

To obtain the value of $m(A_i)$, each VM captures the features of the data flow received. The flow features usually include the types of packets, the addresses of the source hosts and destination hosts, the parameters in the fields of the headers and etc. Each VM then independently analyzes these features using its own intrusion detection algorithms like k-nearest neighbor (KNN), support vector machine (SVM), decision tree (DT) and etc.

We use the value of $m(A_i)$ to express only the possibility of the attacks defined in the set A_i that might contain multiple attacks, but cannot distinguish the possibility of each attack in the subset of A_i, which is quite reasonable when multiple attacks share the similar flow features in the data flow. It is particularly useful in case we cannot distinguish more specific evidence among different types of attacks in some situations.

$$E(VM_k) = \ m_k(A) \tag{15}$$

Sometimes, multiple intrusion detection algorithms may apply concurrently on a single VM in order to better identify the types of the attacks. Each intrusion detection algorithm is able to have individual mass function $m_l(A)$. For this situation, we can evaluate the overall mass function $\overline{m}(A)$ by introducing a weight on every individual mass function $m(A)$.

$$\overline{m}(A) = \frac{\sum w_l \times m_l(A)}{\sum w_l} \tag{16}$$

4.2 Evidence fusion

The evidence is then combined by using the evidence fusion of Dempster-Shafer theory. Dempster's rule for combination gives a numerical procedure for fusing together multiple pieces of evidence from unreliable observers. Evidence Fusion is implemented through two stages.

At the first stage, the VM-based evidence is fused on each pair of the VMs. We suppose two neighboring VMs generate their evidence $E\left(VM_{ik-1,jk-1}\right)$ and $E\left(VM_{ik,jk}\right)$ respectively, as shown in Fig. 3. According to Eq. (7) and Eq. (15):

$$E\left(NS_{k-1,k}\right) = m_{k-1}(A) \oplus m_k(A) \equiv m_{k-1,k}(A)$$

$$= \frac{\sum_{A_i \cap A_j = A} m_{k-1}(A_i) \times m_k(A_j)}{\sum_{A_i \cap A_j \neq \emptyset} m_{k-1}(A_i) \times m_k(A_j)} \quad (17)$$

At the second stage, the cloud-based evidence is fused among all VMs. According to Eq. 8 and Eq. (17):

$$E(NS) = m_{1,2}(A) \oplus m_{2,3}(A) \oplus \ldots \oplus m_{k-1,k}(A) \equiv m_{1,2,\ldots,k}(A)$$

$$= \frac{\sum_{A_i \cap A_j \cap \ldots \cap A_l = A} m_{1,2}(A_i) \times m_{2,3}(A_j) \times \ldots \times m_{k-1,k}(A_l)}{\sum_{A_i \cap A_j \cap .. \cap A_l \neq \emptyset} m_{1,2}(A_i) \times m_{2,3}(A_j) \times \ldots \times m_{k-1,k}(A_l)} \quad (18)$$

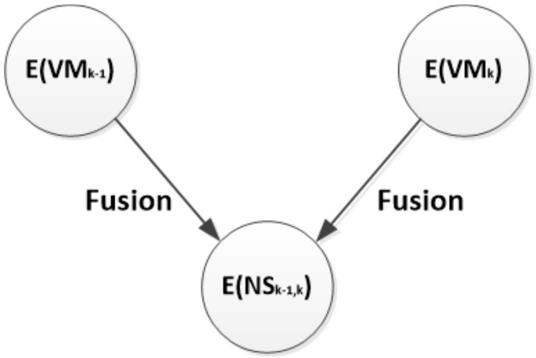

Figure 3: Evidence fusion

4.3 Evidence judgment

Evidence judgment is implemented through two functions. To evaluate the evidence, we define belief function and plausibility function. The belief function represents the weight of evidence supporting one's probability. The plausibility function is the weight of evidence that does not refute this one.

The belief function is the lowest bound of possibility of the malicious attacks detected on the VMs.

$$Bel(A_i) = \sum_{A_j \subseteq A_i} m_{1,2,\ldots,k}(A_j) \quad (19)$$

The plausibility function is the highest bound of possibility of the malicious attacks detected on the VMs.

$$Pls(A_i) = \sum_{A_j \cap A_i \neq \emptyset} m_{1,2,\ldots,k}(A_j) \quad (20)$$

Therefore, the evidence interval is consisted of $Bel(A_i)$ and $Pls(A_i)$. We call $Bel(A_i)$ the lower limit and $Pls(A_i)$ the upper limit. The interval $[Bel(A_i), Pls(A_i)]$ indicates the uncertain of the judgment for the malicious attack A_i, as shown in Fig. 4. A large value of $Pls(A_i) - Bel(A_i)$ indicates the degree of not clear whether the set of attack A_i is true or false.

At last, the evidence judgment function $f(A_i)$ is defined to evaluate the possibility of intrusion, and is indicated as:

$$f(A_i) = Bel(A_i) + \frac{|A|}{|\Omega|} \times \left(Pls(A_i) - Bel(A_i)\right) \quad (21)$$

It shows how probable the attack A_i is.

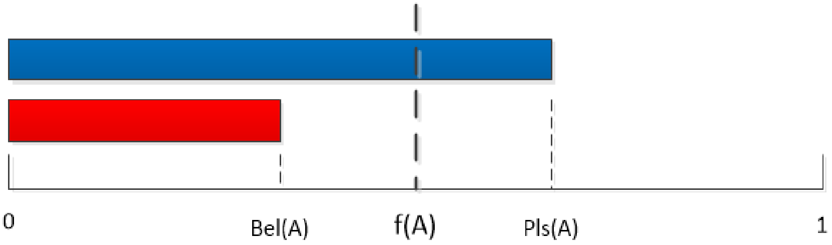

Figure 4: Lower bound and upper bound of evidence judgment

5 Experiment

We designed an experiment scenario to evaluate the performance of our proposed algorithm on the cloud. We built up our proposed multi-VMs IDS by creating 3 instances of VMs (VM_1, VM_2 and VM_3) on the open source cloud platform, OpenStack. An attacker and a normal user from the external network, implemented by two packet generator programs, generate the TCP SYN flood and the HTTP request traffics concurrently to these 3 VMs located in the internal network. The KNN algorithm is used as the default algorithm to determent the initial estimates of the evidence (m_1, m_2 and m_3) of each VM.

During the testing, we collected the network traffic observed on the VMs over a period of time. The statistical tests of KNN are then carried out on the observed traffic to determine whether that behavior is a known attack or not.

Comparing the results between our proposed algorithm of 2-level evidence fusion and the traditional algorithm that uses one way fusion, our results have higher successful rate in classifying the malicious attack as intrusions, because the two-level evidence fusion are introduced to our algorithm for better estimating the probability of an attack on each pair of the VMs. The simulation results are shown in Tab. 1.

Table 1: Simulation results of evidence fusion

Experiment No.	m1, m2, m3	f(A) (Traditional)	f(A) (Proposed)
1	0.7, 0.6, 0.1	0.919	0.968
2	0.2, 0.3, 0.8	0.916	0.941
3	0.6, 0.4, 0.3	0.874	0.924

6 Conclusions

The virtualization nature of the cloud provides the flexibility for deploying network service across different VMs, but this also makes it susceptible to the distributed attack. We proposed a multi-VMs intrusion detection framework, in which each VM observes and analyzes the evidence independently with its own detection algorithm, but makes the collaborative intrusion decision with other VMs. The design of two-level evidence fusion, both VM-based level and Cloud-based level, allows the potential interaction between

VMs, usually neglected by other collaborative models, to be processed now in our model and thus can provide more accurate results.

Acknowledgement: This work was supported by the Macau Science and Technology Development Fund under Grant No. 096/2013/A3 and 0026/2018/A1, and the Macau Fund.

References

Abhaya; Kumar, K. (2016): An efficient network intrusion detection system based on fuzzy C-means and support vector machine. *International Conference on Computer, Electrical & Communication Engineering*, pp. 1-6.

Bezdek, J. C. (1981): *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press.

Chang, Y.; Li, W.; Yang, Z. (2017): Network intrusion detection based on random forest and support vector machine. *IEEE International Conference on Computational Science and Engineering*, vol. 1, pp. 635-638.

Chonka, A.; Xiang, Y.; Zhou, W.; Bonti, A. (2011): Cloud security defense to protect cloud computing against HTTP-DoS and XML-DoS attacks. *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1097-1107.

Dempster, A. P. (1976): Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, vol. 38, no. 2, pp. 325-339.

Ficco, M.; Tasquier, L.; Aversa, R. (2013): Intrusion detection in cloud computing. *IEEE Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 276-283.

Gul, I.; Hussain, M. (2011): Distributed cloud intrusion detection model. *International Journal of Advanced Science & Technology*, vol. 34, pp. 71-82.

Guo, Y.; Liu, F.; Cai, Z.; Xiao, N.; Zhao, Z. (2018): Edge-based efficient search over encrypted data mobile cloud storage. *Sensors*, vol. 18, no. 4, pp. 1189.

Khan, L.; Awad, M.; Thuraisingham, B. (2007): A new intrusion detection system using support vector machines and hierarchical clustering. *VLDB Journal*, vol. 16, no. 4, pp. 507-521.

Khorshed, M.; Shawkat, A.; Saleh, A. W. (2012): A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing. *Future Generation Computer Systems*, vol. 28, no. 6, pp. 833-851.

Mewada, A.; Gedam, P.; Khan, S.; Reddy, M. U. (2010): Network intrusion detection using multiclass support vector machine. *International Conference on ACCTA*, vol. 1, no. 2, pp. 172-175.

Mishra, P.; Pilli, E. S.; Varadharajan, V.; Tupakula, U. (2017): Efficient approaches for intrusion detection in cloud environment. *IEEE International Conference on Computing, Communication and Automation*, pp. 1211-1216.

Modi, C.; Patel, D. R.; Borisaniya, B.; Patel, H.; Patel, A. et al. (2013): Review: A survey of intrusion detection techniques in cloud. *Journal of Network & Computer*

Applications, vol. 36, no. 1, pp. 42-57.

Mukkamala, S.; Janoski, G.; Sung, A. (2002): Intrusion detection using neural networks and support vector machines. *IEEE International Joint Conference on Neural Networks,* vol. 2, pp. 1702-1707.

Singh, D.; Patel, D.; Borisaniya, B.; Modi, C. (2016): Collaborative IDS framework for cloud. *International Journal of Network Security*, vol. 18, no. 4, pp. 699-709.

Shafer, G. (1976): *A Mathematical Theory of Evidence*. Princeton University Press.

Vaid, C.; Verma, H. K. (2015): Anomaly-based IDS implementation in cloud environment using BOAT algorithm. *IEEE International Conference on Reliability, Infocom Technologies and Optimization*, pp. 1-6.