International Journal of Distributed and Cloud Computing
7 (1) June 2019, 30-35
http://www.publishingindia.com/ijdcc

# Cloud Security Framework: VM Features Based Intrusion Detection System

Yakuta Tayyebi[1*] and D. S. Bhilare[2]
[1]Department of Computer Science, I.L.V.A. Commerce & Science College, Indore, Madhya Pradesh, India.
Email: tayyebiyakuta@gmail.com
[2]Computer Center, D.A.V.V, Indore, Madhya Pradesh, India. Email: bhilare@hotmail.com
*Corresponding Author

**Abstract: Cloud services provide resources that are accessed remotely over the network. The distributed architecture of Cloud deployed over the internet, exposes it to several network attacks. To provide a secure architecture, several frameworks using various security tools have been proposed. However, due to the dynamic nature of cloud infrastructure, newer challenges have to be addressed. Here, we propose a Multilevel Intrusion detection system using virtual machine's feature and behaviour to provide a secure Cloud architecture. This framework deploys VM's feature based signature intrusion detection system on each VM (instance) in the cloud. IDS are configured for every VM at the time of its launch according to the features defined by the user and updated thereafter according to the traffic pattern at that VM, by a Control unit at the host level. The framework developed works at optimal computational cost, minimum packet drop and acceptable attack detection rate. For verifying the functional validation and effectiveness of this framework, we have developed a prototype considering few known attacks signatures.**

**Keywords: Cloud computing, Network based intrusion detection system, Snort, Virtual machine.**

## I. Introduction

Cloud Computing provides configurable resources that can be accessed over the internet. Recourses from shared pool are rapidly allocated and reallocated as per the requirements with minimum management [1]. The Cloud computing infrastructure uses virtualization technologies that dynamically allocate Virtual Machines (VM). These instances are interconnected through a virtual private network. Virtualization software like VMware and Virtual Box are available for providing virtual machines over host machine.

Due to shared infrastructure accessed over the internet, Cloud architectures are highly prone to network attacks. According to a survey done by International Data Corporation, security is one of the top concerns of the organizations in adoption of Cloud [2]. Researchers [3] have analyzed issues of Cloud security at various levels of cloud architecture. It has been concluded in a survey, that network intrusion is one of the top three security concern in cloud architecture [4]. Attacks on cloud infrastructure are mainly classified into two security threats, i.e. through an external attacker and through Insider attacker.

Malicious Insider can be an employee at user side or an attacker at the Cloud service provider side. An attacker having access over a VM can fire critical attacks on other VM's on the same host, compromising their functionality and integrity. Conventional security tools like honeypot and Intrusion Detection System (IDS) can deal with advance security issues when used for Cloud architecture. Heavy network traffic because of large number of VMs in the Cloud has to be considered while deploying IDS. VM instances are allocated and reallocated dynamically in cloud which requires the IDS to scale accordingly.

This paper elaborates working of the proposed security framework; Multilevel VM feature based Intrusion Detection System to secure Cloud. IDS have been configured at every virtual machine in Cloud. These detection systems are deployed using Snort configured with signatures specific to VM's feature. Network traffic on each virtual machine is monitored in inline mode. The packets are captured and sent to the Control Unit for detailed offline analysis and VM's profile is updated accordingly.

The paper is structured as follows: Section II presents surveys of related work done in Cloud. The overall working of the proposed framework is elaborated in Section III. Deployment and analysis of the framework are discussed in Section IV. Section V provides the conclusion, followed by references.

## II. Literature Review

An Intrusion Detection System (IDS) is used to detect an unauthorized user accessing the resources or an authorized user misusing its privileges. IDS monitor the system for suspicious activities such as unauthenticated access to the resources; attempt to gain extra resources, eavesdropping the traffic etc. Intrusion Detection Systems are categorized on the basis of

the detection technique they use (signature based or anomaly based) or by the place where they are deployed and type of data they monitor in the system. The broad category of IDS according to the type of data captured are host based IDS and network based IDS.

### *A. Related Work*

This section provides a brief account of the work done to secure cloud structure with NIDS. Various proposed frameworks using IDS in cloud architecture at different locations for providing security have been discussed [5-11]. The challenges in deploying IDS to secure cloud include scalability, robustness, cost-effective and systems compatible with high cloud traffic. Cloud service providers and users are interested in solutions that are dynamically scalable at an acceptable cost [4]. We will also discuss the limitations of the existing systems and their effectiveness.

A cooperative IDS framework for Cloud computing networks [5] consist of multiple IDS monitoring data in their respective region, working in cooperation and communicates alert to each other to detect the DDoS attack.

Mazzariello *et al.* [6] proposed implementing IDS into an open source (Eucalyptus) Cloud platform. Snort is deployed at the Cloud controller and host (hosting VMs). This deployment resolves the issue of deploying IDS at each instance and provides a cost-effective solution for known attacks.

Bakshi *et al.* [7] have proposed installing IDS at the virtual switch to identify distributed attack. The traffic at virtual network is captured and stored in the database where they are scanned for the presence of patterns of the known attack to detect intrusions. Packets from specified IP addresses identified as the attacker are dropped. Thus, this system can deal with the distributed attack.

Gul and Hussain [8] proposed using third party services to secure Cloud architecture. IDS are deployed at the front end and alerts log are sent to a third party for analysis purpose. This third party issues a detailed report to user community and Cloud service provider. However there is a no consideration made for detecting attacks at the virtual network level.

Smallwood and Vance [9] proposed detection through deep packet analysis. The concept of deep packet inspection for network attack will demand a large number of resources in Cloud, which makes the concept not efficient for real-time implementations.

Modi and Patel [10] have proposed a hybrid network intrusion detection system framework to detect attacks. Bayesian classifiers, Associative classifiers, and Decision tree classifiers are used along with signature detection techniques. But the processing cost of this framework is high as the packets are scanned using multiple classifier techniques.

Researchers [11] have deployed a framework to mitigate network attacks in cloud architecture. DOS, session hijacking, DNS attack, and Eavesdropping are considered.

Lambert *et al.* [12] emphasis on evaluating IDS techniques on the basis of processing cost incurred. A highly accurate NIDS is considered noneffective if cost of monitoring data is high as the packet loss incurred increases the probability of an attack not being detected.

### *B. Problem Statment*

Existing solutions perform intrusion detection by scanning the packets for all attack signatures, making the process consume large amount of resources leading to possibility of packet drop at high cloud traffic. The analysis is done without considering the VM's features and behaviour. There is no concept of selection of attack signature that can effectively cut down the detection time and resources required in the cloud architecture. Cost effective and lightweight IDS sensors should be deployed on each VM using VM's features which will be more robust. Further, a detailed analysis can be done at the host machine level by a control unit. Control unit also configures and updates the IDS according to the VM characteristics and behavior.

## III. Overall Working of Proposed Framework

We have proposed a framework to detect network attacks in cloud architecture which addresses the gaps identified in previous techniques. The proposed framework VM feature based NIDS will be effective in identifying attacks from other VM (insiders) on the host machine and external (outside attackers) as well. Detail designed and deployment of the framework is discussed that work in real time.

The motivation of using VM feature based system is to enhance the attack detection capacity of each IDS sensor deployed at the VM network interface by using lightweight signature-based detection tool (snort) with selected signatures. This system emphasizes at catering real time security requirements of cloud system at optimal resource utilization and compatible with high VM traffic (no packet dropping).

NIDS sensors are configured on each VM's network interface in Cloud with selected signatures, running in inline (intrusion prevention) mode. IDS inspecting traffic at the virtual network of the system should be compatible with the high Cloud traffic. These signatures are selected by the Control unit on the basis of VM's features initially. The Control unit creates a profile for each VM at the time of its launch. The initial VM profile is created on the basis of VM features at time of launch of the VM. The Control unit has a description of all the known attack signatures for reference.

Each VM profile consists of weights for known signatures for detecting the attacks that particular VM can possibly suffer and launch in cloud architecture. Initially, weights are attached on the basis of attack severity and VM's feature. The signatures with higher weights are given priority by the IDPS deployed at VM network interface working in inline mode. Scanning the packets for a limited number of signatures at VM prevents

packet dropping in case of high traffic. Also IDS will have lower false positive as the signatures of the frequent, and sever attacks are used.

The packets at host are captured are sent to the Central unit for a detailed offline analysis of the known attack signature with lower weight values. If a new attack is detected, updates are made in its profile accordingly. The weight for rule for that attack is updated in VM profile. The signature set to be used at the VM's interface is updated periodically with the latest profile in which the weights are according to the severity and frequency of occurrence of attacks. The Control unit is connected to the central unit responsible for anomaly based detection to find unknown or new attacks. Any new attacks discovered by the central unit are communicated to Control unit. Weights are attached to these newly discovered attacks for each VM and are included in each VM profile DB respectively. The signatures with higher weights in the VM's profile are considered for detection by the IDPS sensor deployed at the VM virtual network interface making it fast, cost-effective, compatible with high VM traffic and suitable for cloud infrastructure. The weights for signatures are determined in the profile definition phase when the VM is launched by the user and is updated periodically as VM works in its normal behavior.

The signature based detection tool is applied at the VM interface level. Snort [13] a freely available open source NIDS tool is used to detect and prevent attacks. It scans the network traffic, pre-processes packets and compares traffic packet with the patterns of known attack signatures to detect intrusions if any. The alerts are sent to service provider and customer.

Snort is a lightweight Network Intrusion Detection System (NIDS) that can also work as a packet sniffer and logger system. Snort's real-time alerting capability makes it suitable to be deployed at VM level in cloud infrastructure. Snort uses libpcap for packet capturing.

Thus, the Multileveled profile based system will be capable of monitor multiple VMs, and to detect and prevent external as well as internal intrusions. The basic model of the proposed framework is shown in Fig. 1.

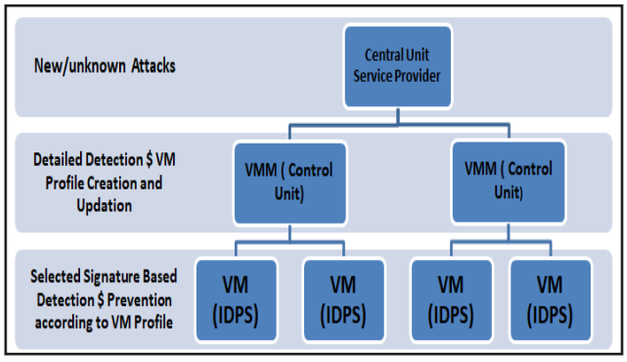


Fig. 1: Multileveled IDPS

The detailed working of the system can be classified in phases described below:

### *A. Initial Profile Definition*

When new VM is launched by the users in the cloud, profile for that VM is defined on the basis of its features. Some of the parameters of the VM profile are determined by the characteristics of the VM as defined by the user like OS, hardware requirement, Software requirements, purpose etc.

During the profile definition phase, the network interface of the newly allocated VM is scanned for all the known attack signatures for a particular time period (Initial profile definition phase) that can be configured by the administrator. The Central unit scans the alert log generated for analysis and profile generation.

*Algorithm: Weight Calculation*

*Input:* VM Features as provided by the user
Alert log generated for profile generation

*Output:* Weights for each known attack signatures

1: Start

2: for each attack signature i do

3: Calculate weight

$$W_i = \left[ \left[ (Fi)^* (£) \right] + \left[ (Si) \right] \right] * (Ai)$$

4: return

Where:

Wi = Weight for signature i

*F*i = Frequency of Attack pattern i in Alert Log

*A*i = Applicability of signature with respect to VM (1/0)

£ = 1/ Total No. of signatures to be configured

*S*i= Severity Score for attack i provided by service provider

Weights for the known signature are determined specific to the VM on the basis of the frequency of attack, the severity **o**f the attack and applicability of attack with respect to that VM in the normal working period. Severity score of an attack will be calculated using Common Vulnerability Scoring System (CVSS). It is used for determining the severity of system security issue. CVSS assign severity ratings to all the known attack signatures. All the universal known signatures are present in the signature database at Control unit.

IDS consume high CPU bandwidth if traffic is checked for all the known attack which results in packet drop at high traffic. Enabling only limited rules specific to the system being monitored will improve the performance. For example, if the VM launched has windows OS then the rules specific to windows should be included; if VM is a database server rule related to it should be included. Applicability with respect to VM (μ) will have the value 1 if the attack signature is relevant to the VM configuration else it will be 0 as shown in Table I.

Table I: Applicability Criteria

| Applicability Criteria | Value of $\mu$ |
|---|---|
| Attack Signature Applicable for a VM | 1 |
| Attack Signature Not Applicable for a VM | 0 |

The VM profile will contain a weight for each known signature and the signatures defining the abnormal behavior of that VM. MY SQL is used to maintain the VM profile. Signature based NIDS Snort is used to scan the traffic at VM interface and traffic packets at Control unit.

### *B. Intrusion Detection and Prevention at VM Network Interface (Inline Mode)*

Once the profile for the newly launched VM is developed in the Initial Profile definition phase, the signature based IDPS at the VM's network Interface is configured. Attack signatures with a weight above a certain threshold are considered while scanning the traffic at the VM sensor in real time. The IDPS (inline mode) deployed over the VM interface are configured to scan traffic for specified signatures as determined by the weights of the signature in the VM profile. Also, the signatures specific to VMs behavior are added to the configuration file. IDPS scans the traffic at the VM interface only for attack signature having a weight above a particular threshold in that VM's profiles (and not for the entire set of known signatures). This makes the IDPS work efficiently without any packet drop at high traffic and low computational resource consumption. Snort is capable to perform real-time traffic scanning on the network protocols. The packets with the pattern of a severe attack that can affect the working of the VM will be dropped by bridge interface by the IDPS working in inline mode. Alerts generated for the attacks detected in this phase and send to the Alert log Database. Alert log database will further sent alert messages to Users and cloud service provider as per requirements.

### *C. Detailed Detection at the Control Unit and Central Unit*

The packets of traffic obtained from the interface are sent to the Control unit for detailed analysis. The time interval for capturing packets in a file and sending it for analysis can be configured by the service provider depending on the network traffic at VM and workload of the Control unit. The packet file is scanned for detailed detection (IDS working in offline mode) against the known attack signatures with weights lower than a threshold value which will be decided by the service provider on the basis of traffic load at VM and user category. Alerts generated for the attacks detected in this phase and send to the alert log database for further investigation. Packets are sent to the central unit for behavior analysis to detect any unknown attacks and abnormal traffic patterns. Alerts are generated for the anomaly detected in this phase and send to the alert log database for further investigation. Signatures are created for the new / unknown attacks according to the traffic behavior analysis and are added to the signature database at the Central unit.

### *D. Updating of VM Profile at Control Unit*

The VM profile is updated according to the attack detected for that VM. The alert log of the attack generated is analyzed and the weight of the signatures of the recently detected attack for a VM in increased in its profile to improve the detection accuracy at the VM sensor.

*Algorithm: Profile Weight Update*

*Input:* Alert log for the known attack signatures
Wo Old Weights for the Known attack signatures
Severity score and Applicability for New Signature

*Output*: Updated Weights for known attack signatures
Weights for new attack signatures

1: Start
2: for each attack signature
3: if (Signature i already exist)
4: Calculate update weight

$$W_{Ni} = \left[\left[ Fi * £ \right] + \left[ WOi \right]\right] * Ai$$

if
5: end if
6: else (if signature i of new attack)
7: Calculate Weight for New Signature

$$W_i = \left[\left[ Fi * £ \right] + \left[ Si \right]\right] * Ai$$

if else
8: end if else
9: end for
10: return

Where:
WNi = Updated Weight for signature i
*F*i = Frequency of Attack pattern i in Alert Log
*A*i = Applicability of signature with respect to VM (1/0)
£ = 1/ Total No. of signatures to be configured
*Si* = Severity Score for new attack i provided by service provider

The frequently detected attack patterns will be given priority as they have higher weights as compared to other rules in the profile of that VM. Signature database is updated frequently with newly discovered unknown attack signatures from the Central Unit. The Control unit updates the VM profile according to network traffic behavior at the VM interface. The updated VM profile for each VM is used to update the configuration of IDPS at the VM network bridge interface. The weights attached to the signature are dynamic in nature and changes with every alert log scan cycle. Overall working of the framework is shown in Fig. 2.

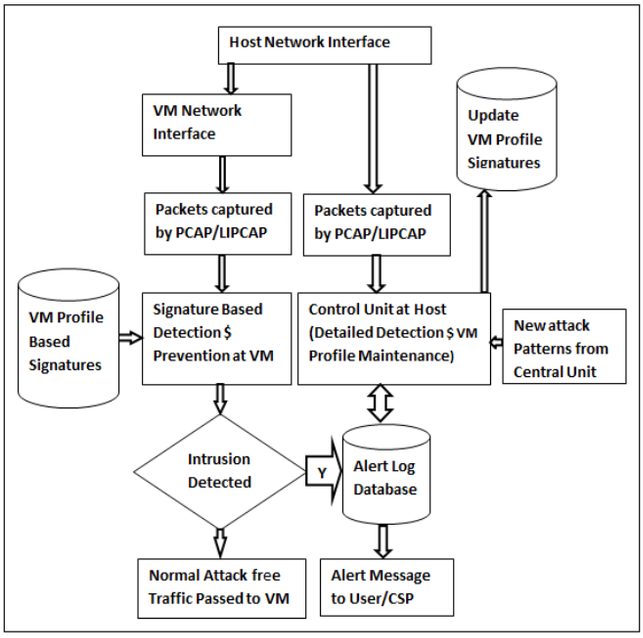


Fig. 2: Traffic Packet Flow

## IV. Experimental Setup and Analysis

A prototype of the proposed framework has been set up with an open source private cloud using few known attack signatures. We have evaluated the system in terms of attacks it can detect and prevent, with no packet drop and optimal system resources consumed.

### *A. Experimental Setup*

Instances are launched having different operating systems and configuration using the VMWARE hypervisor. The IDPS at each VM is configured according to the VM feature and behaviour as generated by the Central unit at VMM level. We have installed Signature Based IDS Snort [13] on each VM. Many freely available rulesets, like Vulnerability Research Team (VRT) ruleset and the Emerging Threats (ET) ruleset can be used. IDS are configured to use signatures from these rulesets. Snort uses a set of signatures written in a configuration file according to the VM feature. Traffic is captured using utilities Net: pcap and libpcap. The alert logs are generated in unified2 format which are then send to control unit. Barnyard2 tool is used to log the alerts to the alert database. The attack signatures for denial of service and Port Scan attacks have been considered. Hping [14] and nmap [15] are used to fire DOS and Port Scan attack on VMs. The NIDS at VM interface inspects network traffic for attacks and prevents the malicious traffic. The traffic at the host machine is captured and sent to the Control unit for detailed analysis and signature weight updating. Control unit is deployed at a privileged cloud service monitor VM or host machine level. BASE (Basic Analysis and Security Engine) a GUI interface is used to analyse the alerts.

### *B. Analysis*

This is a prototype implementation of the concept. We are considering few known attack signatures for our implementation. From this set of known signatures, selections are made according to the proposed algorithm and the selected signatures are used to configure the IDPS at VM's virtual network interface. The quality of a NIDS should be evaluated on the basis of the genuine attacks detected by it against the number of false alerts generated [16]. The analysis is done on the basis of the following factors:

- Adaptablity: The service provider updates the Control unit with signatures for newly discovered attacks. After analysis, weights are calculated for new rules and added into Snort configuration file. Also the weights are updated to reflect any change in VM feature or profile. This makes the proposed framework scalable and adaptable.
- Computational Cost: The IDPS deployed at each VM interface works with a limited number of signatures at optimal computational cost. Fig. 3 shows the comparison of resources required at the VM in case of VM features based IDS configuration and IDS configured with all signatures. The graph shows that the IDS configured with selected signature on the basis of weight consumes less resourses than the case when the IDS is configured with all the signatuter. There is an drop of 10 to 15 percent in the resources consumption at VM level, depenting upon the no. of packet processed.

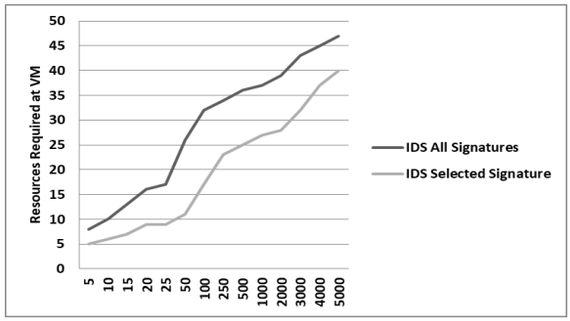


Fig. 3: Comparative Graph of VM Level Resource Required

- Detection Accuracy: IDPS are deployed at VM interface with specific signatures. Accuracy persentage is calculated as per the formula.

$$Accuracy = ((TP+TN) / (TP+TN+FP+FP) * 100)$$

*Where:* $TP$ = True Positive Cases

$TN$ = True Negative Cases

$FP$ = False Positive Cases

$FN$ = False Negative Cases

Fig. 4 shows the comparison of accuracy level attained by IDS when entire set of signatures were used to the level of accuracy attained when selected signatures were used according to the VM's profile. IDS when configured with

selected signature shows an improvement in accuracy present. This is due to reduction in false positive cases, as the feature and profile of VM are considered by the signature weight calculation algorithm.

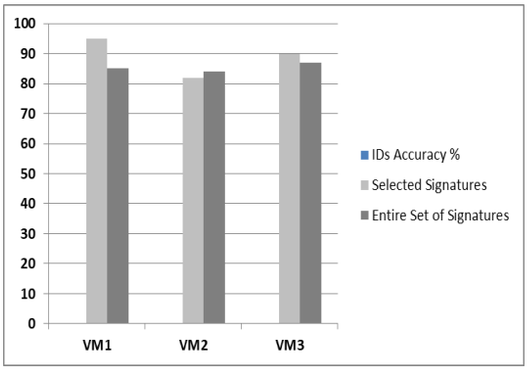


Fig. 4: IDS Accuracy in %

- Low Packet Dropping at VM Interface: The IDPS at VM interface is configured with selected signatures according to the VM feature and behaviour. The system works efficiently at high traffic without packet drops.
- Compatibility: Proposed VM level NIDS framework is compatible with any communication protocol (TCP/IP, UDP) and can be deployed on window and Linux based VMs in cloud architecture.

The analysis shows that VM feature based IDPS increases robustness and is cost effective at acceptable detection rate for cloud architecture.

## V. Conclusion

Ensuring security in Cloud Infrastructure has lead researchers to investigate the applicability of traditional security techniques and development of new techniques. Virtualization and shared resources in Cloud brings additional challenges in deploying IDS to secure cloud. We have critically analyzed working of IDS in Cloud and focused on key features required of IDS to provide security in Cloud. This paper provides an overview of working of VM feature and behaviour based NIDS for identifying intrusions in cloud. The concept of scanning the VM traffic for specifically selected signatures according to the VM feature is introduced. The VM profile contains the weights for known signatures which will ensure efficient attack detection at VM interface without packet drop in case of high traffic. The proposed system satisfies all the Cloud network security requirements.

## References

[1] International Data Corporation. Retrieved from http://blogs.idc.com/ie/wp-content/uploads/2009/12/idc_cloud_challenges_2009.

[2] F. Gens, IT Cloud services user survey, top benefits and challenges, International Data Corporation.

[3] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and R. Muttukrishnan, "A survey on security issues and solutions at different layers of cloud computing," *The Journal of Supercomputing*, vol. 63, no. 2, pp. 561-592, 2013.

[4] L. Martin, "Awareness, trust and security to shape government cloud adoption," White Paper, 2010.

[5] 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 (ICPPW),* pp. 280-284, 2010.

[6] C. Mazzariello, R. Bifulco, and R. Canonico, "Integrating a network IDS into an open source cloud computing environment," in *Proceedings of the Sixth International Conference on Information Assurance and Security*, pp. 265-270, 2010.

[7] A. Bakshi, and B. Yogesh, "Securing cloud from DDOS attacks using intrusion detection system in virtual machine," in *Proceedings of the Second International Conference on Communication Software and Networks,* pp. 260-264, 2010.

[8] I. Gul, and M. Hussain, "Distributed cloud intrusion detection model," in *International Journal of Advanced Science and Technology*, pp. 71-82, 2011.

[9] D. Smallwood, and A. Vance, "Intrusion analysis with deep packet inspection: Increasing efficiency of packet based investigations," in *Proceedings of the International Conference on Cloud and Service Computing*, pp. 342-347, 2011.

[10] C. N. Modi, and D. Patel, "A novel Hybrid-Network Intrusion Detection System (H-NIDS) in cloud computing," *2013 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, Singapore, pp. 23-30, 2013.

[11] S. Gupta, P. Kumar, and A. Abraham, "A profile based network intrusion detection and prevention system for securing cloud environment," *International Journal of Distribute Sensor Networks*, pp. 1-12, 2013.

[12] L. Schaelicke, T. Slabach, B. Moore, and C. Freeland, "Characterizing the performance of network intrusion detection sensors," in *RAID 2003, LNCS 2820*, pp. 155-172, 2003.

[13] Snort Tool, Snort-home page. Retrieved from https://www.snort.org/, May 28, 2018.

[14] Hping security tool. Retrieved from http://www. hping. Org/, June 10, 2018.

[15] Nmap. Retrieved from http://nmap.org/, June 10, 2018.

[16] Y. Tayyebi, and D. S. Bhilare, "Cloud security through Intrusion Detection System (IDS): Review of existing solutions," *International Journal of Emerging Trends & Technology in Computer Science*, vol. 4, no. 6, pp. 213-215, 2015.