2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress

Detecting Denial of Service Attacks in the Cloud

Raneel Kumar
University of the South Pacific
Suva, Fiji
e-mail: raneel.kumar@usp.ac.fj

Sunil Pranit Lal
Massey University
Palmerton North, New Zealand
e-mail: s.lal@massey.ac.nz

Alok Sharma
University of the South Pacific
Suva, Fiji
e-mail: alok.fj@gmail.com

***Abstract*— In this paper, an approach to protecting virtual machines (VMs) against denial of service (DoS) attacks in a cloud environment is proposed. An open source cloud computing platform (Eucalyptus) has been deployed, and experimentation was carried out on this setup. We investigate attacks emanating from one or more virtual machines (VMs) to another VM in a multi-tenancy cloud environment. Various types of DoS attacks are mounted on a webserver VM. To detect such attacks from a cloud provider's perspective, an intrusion detection system (IDS) is needed. In this research we propose and implement an IDS which incorporates a packet sniffer, feature extractor and a classifier as part of its design. We have experimented with the one-class support vector machines (SVM) algorithm for classification of the attacks. The dataset containing time-based traffic flow features is passed through the classifier to detect the attack traffic from legitimate traffic. The proposed IDS design shows promising results in being able to detect the ICMP Flood, Ping-of-Death, UDP Flood, TCP SYN Flood, TCP LAND and DNS Flood attacks with high classification accuracies.**

***Index Terms*— Eucalyptus Cloud, Denial of Service, Intrusion Detection System, One-Class Support Vector Machines, Virtual Machines**

I. Introduction

With the advancement of processing, storage, memory and bandwidth, a new model of computing known as cloud computing has emerged. Cloud computing is a technological advancement in providing information technology infrastructure, platform and software as services over the Internet. The cloud computing service model has cloud providers leasing the intended services to the cloud users in an on-demand manner [1]. The cloud users can lease the service from the cloud and release it back to the cloud when the need has been met. Cloud computing is gradually being adopted by organizations as private, public or hybrid clouds. The adoption of cloud computing presents a number of benefits over the traditional datacenter such as improved agility (on-demand, self-service, elastic resources), fast service provisioning, scalability of services, better resource utilization and reduced operational costs. Recent trends favor the cloud computing adoption and show that cloud computing platforms or cloud based datacenters would be processing more workload and traffic than traditional datacenters. Cisco predicts that by 2018, more than three quarters (78%) of workloads will be processed by cloud based datacenters [2]. In addition, the global cloud IP traffic will account for more than three-fourths (76 percent) of total datacenter traffic by 2018.

A survey by International Data Corporation suggests that security issues in cloud computing is the leading challenge in the adoption of cloud computing [3]. For organizations to transition to clouds, it becomes important for cloud providers to assure significant level of security to the clients. Together with existing security mechanisms such as firewalls and intrusion detection systems (IDSs), cloud providers can also have security mechanisms built into the architecture of the cloud in assuring high level of security to the clients [4].

Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are one of the common attacks on the Internet today. DoS attacks aim to exhaust a system's resources such that it compromises its ability to provide the intended service and thus rendering it unavailable. The Cisco 2014 Annual Security report ranks the effects of DoS attacks in the magnitude of high severity [5]. DoS attacks which mainly target websites can also paralyze Internet service providers. For instance, in August 2013, the Chinese government reported that the largest DDoS attack it had ever faced shut down the Internet in China for about four hours. Also in March, 2013, a 300 Gbps DDoS attack known as DNS amplification attack was observed on the Spamhaus website hosted at CloudFlare where around 30,000 DNS open resolvers were utilized to attack the website [6]. DoS attacks can be broadly classified into three categories [7]:

1. Volume based DoS attacks: These affect servers when a high volume of such traffic is directed towards it. Examples are ICMP flood and UDP flood attacks.
2. Protocol based DoS attacks: These use specific Internet protocols to consume the server's resources. Examples are TCP SYN flood attack and Ping-of-death.
3. Application based DoS attacks: These target the weaknesses of the applications in the Internet. It is also known as Application Layer attacks and examples of it are Slowloris attack and DNS amplification attack.

In a multi tenancy cloud environment, the traditional viewpoint of demarcating the attackers as entities on the outside of the infrastructure does not hold relevance. In a cloud environment, an attacker can lease resources from the cloud to launch DoS attacks on other tenants' virtual machines. Like a cancer, the attacker can use the cloud resources to attack the cloud from within. Therefore it becomes extremely important for cloud providers to detect and thwart such DoS attempts in an effort to safeguard the cloud resources and tenants interests.

978-1-5090-4065-0/16 $31.00 © 2016 IEEE
DOI 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.70

II. Related Work

To detect DoS attacks in a network, an Intrusion Detection Systems (IDS) is normally utilized. An IDS is an advanced security mechanism that is deployed on network or hosts to detect malicious activities such as DoS attacks, Trojan horses and Internet worms. An IDS uses either a signature or anomaly based approach in the detection process. Signature based IDS requires a database of known attack signatures and pattern before it can detect attacks. The open source Snort [8] is a common signature based IDS and has been experimented by Lonua et al. [9] and Chung et al. [10] in the cloud environment with DoS attacks. The limitation of using signature based approaches is that the IDS will not be able to detect novel attacks if the attack signature is not present in the IDS database.

Anomaly based IDS on the other hand uses statistical or machine learning methods to detect attacks and have an advantage over signature based IDS as they are capable of detecting new attacks without having information and experience of the attack. Gupta et al. [11] have proposed a profile based Network Intrusion Prevention System (NIPS) for securing the cloud environment. The open source cloud computing platform used by the researchers is OpenNebula [12]. Here the NIPS which is managed by a cloud administrator examines packets originating from and destined to virtual interfaces of separate VMs and monitors it against the VM's profile. An initial VM profile is created by monitoring all traffic that passes to and from the VM. This traffic is compared against an attack signature database and using the attacks and normal behaviors of the traffic a profile is created. The profile can be updated later by the administrator. For experimentation the ICMP flood and TCP SYN flood attacks had been used by the researchers. An alert and response component relayed the necessary information to the administrator. Although a novel approach of having VM profiles is presented for the IDS/IPS, the research work does not consider a diverse range of DoS attacks in determining the effectiveness of the IDS.

Modi et al. [13] present a comprehensive review of intrusion detection techniques for cloud environments. They highlight on the use of data mining and machine learning techniques for an anomaly based intrusion detection system such as artificial neural networks, fuzzy logic, associate rules, support vector machine, genetic algorithm and hybrids of these. Another important factor to consider when using these techniques is the experimentation dataset that is used. A common dataset that has been widely used by researchers is the KDD CUP 1999 intrusion detection dataset , which has been reported to have discrepancies such as imbalance of the dataset and the aging factor of the dataset itself [14]. Singh and Bansal [15] have used NSL KDD dataset - a subset of the KDD CUP 1999 dataset in evaluating the performance of artificial neural network classifiers.

III. Problem Statement

Shea and Liu's [16] work on virtualization showed that performance of VMs under DoS attacks degrade more than on non-virtualized systems with the same amount of resources. As virtualization is at the core of cloud computing, this implies that VMs and associated services in the cloud are more vulnerable to DoS attacks than on standalone non-virtualized systems. In addition, due to the multi-tenancy nature of cloud environments especially for clouds deployed as IaaS models, there may be malicious tenants in the cloud who can cause harm to legitimate tenants. Thus, through this research work, the issue of DoS attacks within the cloud environment is explored. Furthermore, various work in literature has investigated the two-class classification methods for differentiating the malicious attack traffic from legitimate traffic. This approach has a downfall that the classifier would only learn those types of attacks it would have been trained with. Unknown attacks, may not be accurately detectable. Also, considering the discrepancies in the KDD CUP 1999 dataset, a new dataset is needed.

IV. Proposed Approach

An open source cloud computing platform – HP Helion Eucalyptus Cloud is used to create an IaaS cloud for the experimentation. A testbed is created on this cloud platform consisting of virtual machines (VMs) of which some are normal VMs (legitimate tenants) and a few VMs are equipped with DoS attack tools. As cloud computing leverages the virtualization technology [17], the IDS is placed in the physical servers or nodes where the server resources are virtualized. The mechanism will monitor all incoming and outgoing traffic to and from each VM and alert the cloud provider when a denial of service attack is detected.

The proposed design (Fig. 1) gives an overview of the IDS in the cloud platform. The novelty of the approach is the use of a dataset consisting of time-based traffic flow features with a one-class machine learning classifier in the detection process. The IDS is designed to detect DoS attacks on VMs in the cloud by any external or internal machine in the Internet. In this research, all attacks are carried out on VMs by other VMs within the cloud as the focus is to have an IDS in place to allow cloud providers to be able to detect any DoS attack from within the cloud. Appropriate actions can then be taken by cloud providers against the tenants who engage in such malicious activities. The IDS resides in the host operating system of the physical servers where server resources are virtualized via a hypervisor and where the VMs run. The major components of the IDS are the packet sniffer, feature extractor and classifier. An alert mechanism is also integrated and is responsible for relaying any DoS attack detected by the classifier to the administrator for further action. The IDS design does not consider the case when DoS traffic comes from spoofed source IP addresses. IP spoofing can be addressed by other means such as the MIT's Spoofer Project [18], and by blocking spoofed packets with network and firewall configurations [19].

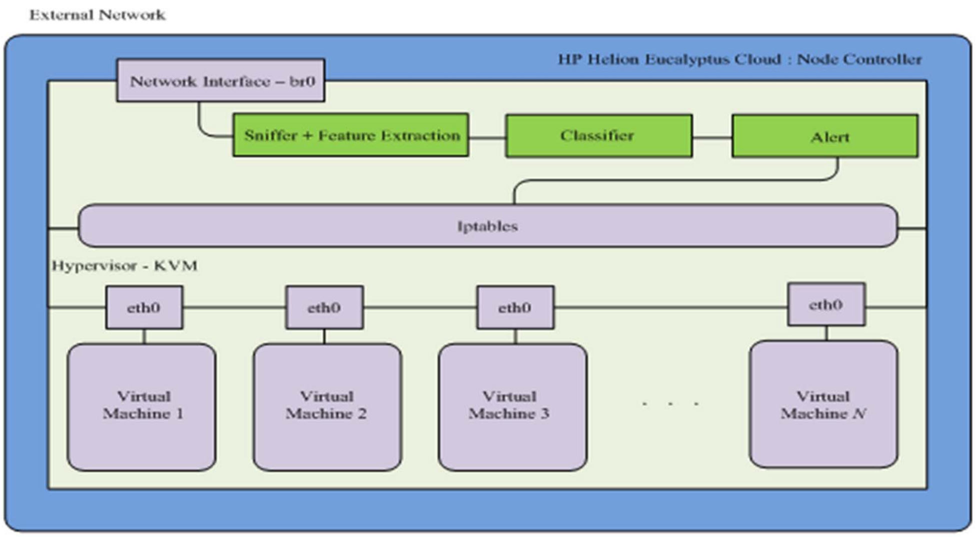


Figure 1: Design of the DoS IDS situated in the Node Controller of the Eucalyptus Cloud

A. *HP Helion Eucalyptus Cloud*

The Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems (Eucalyptus) is an open source cloud platform under the Hewlett-Packard (HP) Helion [20] initiative which provides organizations opportunity to establish IaaS private clouds. Eucalyptus was released in 2008 and has over the years matured into a robust private cloud computing platform. The cloud offers many features such as compatibility with AWS APIs, networking, compute, storage and cloud management. Eucalyptus comprises of five main components and a sixth optional component of which all are stand-alone web-services. The components and the functions of each are provided in Table I [21]. For this research, experimentation has been carried out on Eucalyptus version 4.1.

TABLE I: EUCALYPTUS COMPONENTS AND FUNCTIONALITY

Component	Functionality
Cloud Controller (CLC)	Handle resource arbitration via the CC on the NCs.
Walrus	Stores persistent data in cloud such as machine images and snapshots
Cluster Controller (CC)	Receive requests from the CLC and handle execution of VMs on NCs
Storage Controller (SC)	Provides persistent block storage to the instances and allow creation of snapshots on Walrus
Node Controller (NC)	Manages instance life cycle and send resource availability and utilization information to CC
VMware Broker	Mediates interaction between CC and VMware hypervisor [Optional]

B. *Sniffer and Feature Extraction*

Virtualization on the NC requires a bridged interface namely br0 to be setup. All the local interfaces of the VMs then attach to the bridged interface in order to communicate with the external network. Ideally, for any packets coming through the bridged interface from the external network, it is forwarded to the IPtables firewall within the Linux operating system. The security feature in Eucalyptus which allows VMs to be assigned a security group requires the security group rules to be implemented as firewall rules in the NC's local firewall – IPtables.

The sniffer used for the purpose of this research is a custom built sniffer, developed in C language using the libpcap library for packet capturing on Linux systems. For cloud providers, protecting tenants' privacy is paramount therefore a balance was reached between the volume and nature of data captured to protect the tenants from DoS attacks while at the same time safeguarding their privacy. After careful consideration the data capture had been limited to four IP header fields (Table II), completely leaving the payload untouched.

TABLE II: CAPTURED IP HEADER FIELDS

Source IP Address
Destination IP Address
Bytes of Data Transferred
Protocol (TCP, UDP, ICMP, others)

The IP header fields captured by the packet sniffer is input to the feature extraction script which calculates the selected traffic flow features per incoming IP address (Table III). The traffic flow is divided into 4 sections based on the protocol field of the IP header. The protocols are TCP, UDP, ICMP and Others, which refer to the other types of transport packet being carried over TCP/IP networks. For each protocol, 6 distinct features are calculated and hence there are 24 features in total. These features are then used by the classifier to differentiate between the legitimate and malicious traffic instances. The alert component is a minor component of the DoS IDS. It is responsible for notifying the cloud administrator of DoS attack attempts on the cloud VMs.

TABLE III: TIME-BASED TRAFFIC FLOW FEATURES

Feature	Feature Shortname	Feature description
f_1, f_2, f_3, f_4	Count	Number of occurrence for an incoming IP for each of the protocols (TCP, UDP, ICMP, Others)
f_5, f_6, f_7, f_8	Avg_Count	Average number of incoming IPs for each of the protocols (TCP, UDP, ICMP, Others)
f_9, f_{10}, f_{11}, f_{12}	Bytes_In	Bytes received per incoming IP for each of the protocols (TCP, UDP, ICMP, Others)
f_{13}, f_{14}, f_{15}, f_{16}	Avg_Bytes_In	Average bytes received per incoming IP for each of the protocols (TCP, UDP, ICMP, Others)
f_{17}, f_{18}, f_{19}, f_{20}	Bytes_Out	Bytes sent to the incoming IP for each of the protocols (TCP, UDP, ICMP, Others)
f_{21}, f_{22}, 323, f_{24}	Avg_Bytes_Out	Average bytes sent to incoming IPs for each of the protocols (TCP, UDP, ICMP, Others)

C. One-Class Support Vector Machine

A support vector machine (SVM) is a discriminative classifier where the training points in space are mapped so that the points belonging to the distinct classes are divided by an optimal hyperplane that maximizes the margin between the different classes. The test points are then mapped onto the same space and then predicted to belong to a class based on the side of the margin they fall on [22]. Training data points on the margin or closest to the optimal hyperplane are called the support vectors.

In contrast to traditional SVMs which are capable of prediction in multi-class problems, one-class SVMs attempt to learn a decision boundary that achieves the maximum separation between the points and the origin [23]. A one-class SVM uses an implicit transformation function defined by the kernel function to project the data into a higher dimensional space. This creates a decision boundary (a hyperplane) for majority of the data and collects it in a class. The data outside the decision boundary is considered as outliners [24]. One-class classification is often called outlier or novelty detection because the learning algorithm learns what normal data is and then differentiates any abnormal data from the normal [25]. One-class SVMs have been experimented in a number of classification problems such as document classification [26], image classification [27], and has been tested with a number of standard datasets such as the breast cancer dataset [24]. For the purpose of this research, the libsvm library by Chang and Lin [28] has been used to develop the classier for the IDS. The one-class SVM classifier uses the linear kernel function which is defined by equation 1. The linear kernel, $k(x,v)$ is the result of the dot product of vectors x and v. Vector x denotes instance x_i of the dataset while vector v is the class label, v_i for the corresponding x_i in the dataset. i is an element of {1, 2, 3, … , n} where n is the number of instances in the dataset.

$$k(x,v) = x \,.\, v \qquad (1)$$

Network traffic from the normal client VMs to a webserver VM on the cloud is captured at 5 seconds intervals and traffic flow features are extracted from the traffic for each incoming IP address. This is then used to train the one-class SVM algorithm such that it is able to model the behavior of any legitimate IP address with respect to each IP address's time-based traffic flow features. Once the classifier has been trained it is subjected to test cases which have both traffic from legitimate and malicious IP addresses and the effectiveness of classifier is then measured. Altogether, the sniffer, feature extraction process, one-class SVM classifier and alert mechanism forms a real-time IDS.

V. EXPERIMENTATION

This section covers the details of Eucalyptus cloud setup, the tools needed to simulate active users in the cloud, performing attacks and generating the dataset.

A. Eucalyptus Cloud Deployment

The cloud is deployed with Eucalyptus version 4.1 software. It runs on 10 physical servers and has a dedicated Storage Area Network (SAN) resulting in a computation capacity of 120 vCPUs and storage capacity of 7.3 TB. Each of the servers has the minimal version of CentOS 6.6 installed and the cloud components are installed on top of the operating system. The deployment of each Eucalyptus component is onto individual physical servers which provides these components dedicated server resources to carry out the functions. The SAN is used as a form of block storage for the SC. The cloud uses the Kernel Virtual Machine (KVM) hypervisor to virtualize the computing, network and storage resources and deliver on-demand IaaS. Details of installing Eucalyptus cloud can be found in the Eucalyptus official installation guide [21].

Eucalyptus cloud has to be configured to allow components to communicate with each other. The VMs once launched should be able to communicate with each other and the external network with a certain level of security in place. The *Edge networking* mode which is used in this setup removes the need to place a single Linux server in the data path for all VMs running in a single cluster and additionally removes the need to configure the underlying network to allow passing of VLAN tagged packets in the cloud [21]. A pool of private and public IP addresses is also required to be allocated to Eucalyptus.

B. Testbed

When the VMs are initiated in the testbed, different roles are assigned to VMs to show the different types of tenants leasing the cloud resources. The roles are of 3 types: normal, attack and target. The normal users are tenants with distinct VMs who have the VM for some specific purpose and at the same time communicate with the target machine. The target VM is a VM which hosts a webserver in the cloud. The VMs assigned the attacker role are those VMs that are equipped with DoS attack tools and these are used to carry out the 7 DoS attacks on the target VM. These DoS attacks are ICMP flood, Ping-of-Death, UDP flood, TCP SYN flood, TCP LAND attack, DNS flood and Slowloris. The attack VMs try to disrupt the target VM such that it is not able to provide service to the legitimate normal users.

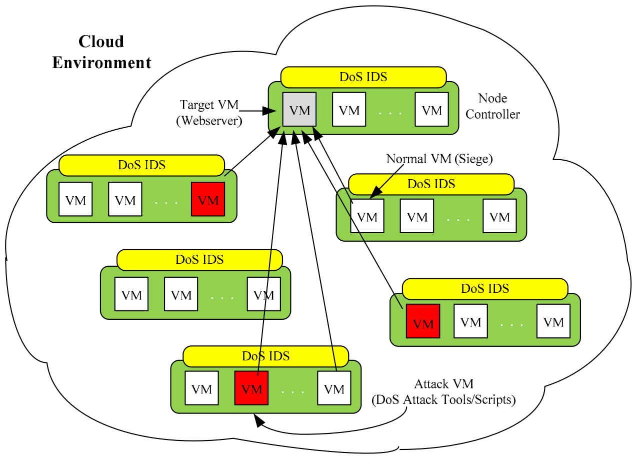


Figure 2: Testbed Setup in the Cloud

As the research is based on VM to VM communication and attacks, it is necessary to simulate an active cloud environment where there are necessary VM to VM communication. As shown in Fig. 2, three types of VMs are setup in the cloud environment to perform specific tasks. Firstly, 15 m1.small (256MB memory, 1 vCPU, 5 GB disk) instances of CentOS 7 and Ubuntu 14.04 are setup as normal VMs whose task is to periodically request for webpages from the webserver VM similar to ordinary Internet use. This is achieved using Siege [29] which is a webserver load test and benchmarking tool and is used to simulate active and legitimate users of the webserver. The target VM is an m3.2xlarge (4GB memory, 4 vCPU, 30 GB disk) instance which runs an Apache 2.4.6 webserver hosting a website of 12 webpages. Finally, 3 m1.xlarge (1GB memory, 2 vCPU, 10 GB disk) instances are setup as attack VMs whose task is to perform the 7 DoS attacks at predefined defined times and scenarios. The m1.small, m1.xlarge, and m3.2xlarge instances are standard VM instance types defined for Eucalyptus cloud VM instances where the number of CPUs, the size of memory, and the size of storage is given to an instance when it boots.

C. DoS Attack Simulation

Seven types of DoS attacks are setup on the 3 attack VMs, which launch attacks in accordance with the predefined scenarios.. The attacks are the ICMP flood, Ping-of-Death, UDP flood, TCP SYN flood, TCP LAND, DNS flood and Slowloris. These attacks target the network, transport and application layers of the TCP/IP protocol stack. Attacks on the network and transport layers are mostly voluminous in nature while application layer attacks target the specific application layer protocols on the victim machine such as the HTTP (Web service). The attacks used in the experimentation cover diverse range of DoS attacks on the Internet, and are categorized as volume, protocol and application based attacks. Through the experimentation the overall effectiveness of the IDS has been tested. These attacks were carried out with a network security tool Hping3 [30], network utility tool Ping and using Slowloris attack script.

The simulation had four scenarios, namely; normal, attack scenario 1, attack scenario 2 and attack scenario 3. The normal scenario had the normal VMs sending legitimate requests to the target VM. In the attack scenarios, the attack VMs carried out DoS attacks on the target VM while the normal VMs sent legitimate requests to the target VM. The attack scenarios had different number of attacks VMs as follows:

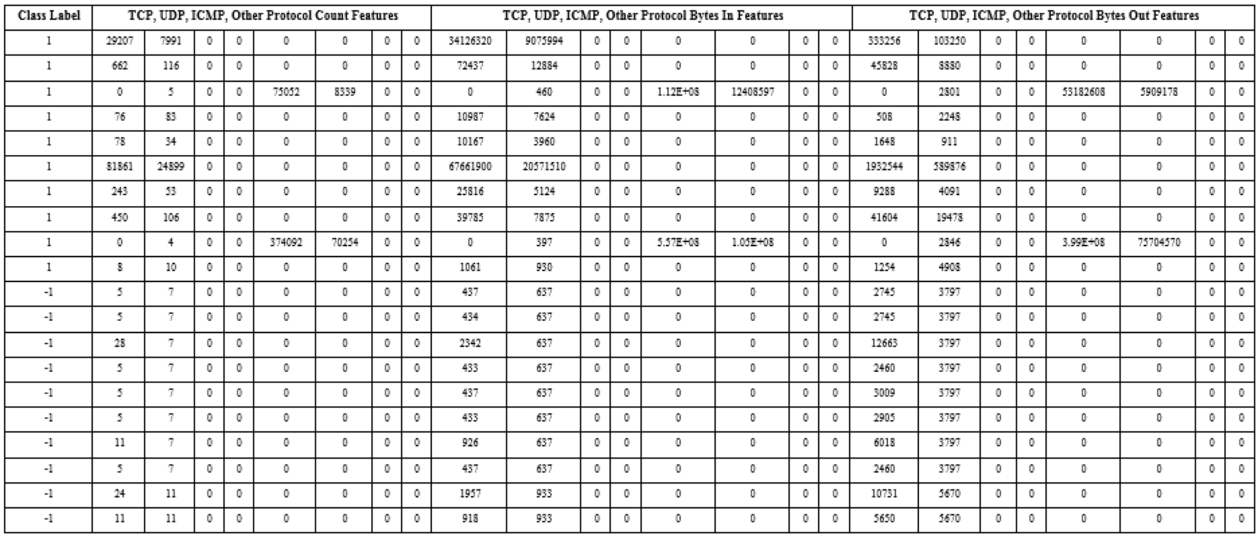
- Attack Scenario 1: An attack VM carries out DoS attacks on the target VM.
- Attack Scenario 2: Two attack VMs simultaneously attack the target VM.
- Attack Scenario 3: Three attack VMs simultaneously attack the target VM.

VI. Dataset

The resulting outcome of the experiment simulation yielded a dataset of 5274 instances each with 24 features. From the 5274 instances, 4592 instances were from user VMs while 682 instances were from attack VMs. The difference in the two classes of legitimate and malicious traffic from the VMs is due to the fact that there were 15 VMs sending legitimate traffic while only 3 VMs were sending malicious traffic.

The original data was split into the training and test dataset groups of an approximate ratio of 3:1. After splitting, the training dataset consists of 3910 instances which belong to the legitimate class only. The 1364 instances belonging to the test dataset consists of 682 instances from the legitimate class and 682 instances from the malicious class. The 682 instances of the malicious class comprises of the malicious instances from the 7 DoS attacks for the 3 attack scenarios.

TABLE IV: DATASET EXTRACT



Class Label	TCP, UDP, ICMP, Other Protocol Count Features								TCP, UDP, ICMP, Other Protocol Bytes In Features								TCP, UDP, ICMP, Other Protocol Bytes Out Features							
1	29207	7991	0	0	0	0	0	0	34126320	9075994	0	0	0	0	0	0	333256	103250	0	0	0	0	0	0
1	662	116	0	0	0	0	0	0	72437	12884	0	0	0	0	0	0	45828	8880	0	0	0	0	0	0
1	0	5	0	0	75052	8339	0	0	0	460	0	0	1.12E+08	12408597	0	0	0	2801	0	0	53182608	5909178	0	0
1	76	83	0	0	0	0	0	0	10987	7624	0	0	0	0	0	0	508	2248	0	0	0	0	0	0
1	78	34	0	0	0	0	0	0	10167	3960	0	0	0	0	0	0	1648	911	0	0	0	0	0	0
1	81861	24899	0	0	0	0	0	0	67661900	20571510	0	0	0	0	0	0	1932544	589876	0	0	0	0	0	0
1	243	53	0	0	0	0	0	0	25816	5124	0	0	0	0	0	0	9288	4091	0	0	0	0	0	0
1	450	106	0	0	0	0	0	0	39785	7875	0	0	0	0	0	0	41604	19478	0	0	0	0	0	0
1	0	4	0	0	374092	70254	0	0	0	397	0	0	5.57E+08	1.05E+08	0	0	0	2846	0	0	3.99E+08	75704570	0	0
1	8	10	0	0	0	0	0	0	1061	930	0	0	0	0	0	0	1254	4908	0	0	0	0	0	0
-1	5	7	0	0	0	0	0	0	437	637	0	0	0	0	0	0	2745	3797	0	0	0	0	0	0
-1	5	7	0	0	0	0	0	0	434	637	0	0	0	0	0	0	2745	3797	0	0	0	0	0	0
-1	28	7	0	0	0	0	0	0	2342	637	0	0	0	0	0	0	12663	3797	0	0	0	0	0	0
-1	5	7	0	0	0	0	0	0	433	637	0	0	0	0	0	0	2460	3797	0	0	0	0	0	0
-1	5	7	0	0	0	0	0	0	437	637	0	0	0	0	0	0	3009	3797	0	0	0	0	0	0
-1	5	7	0	0	0	0	0	0	433	637	0	0	0	0	0	0	2905	3797	0	0	0	0	0	0
-1	11	7	0	0	0	0	0	0	926	637	0	0	0	0	0	0	6018	3797	0	0	0	0	0	0
-1	5	7	0	0	0	0	0	0	437	637	0	0	0	0	0	0	2460	3797	0	0	0	0	0	0
-1	24	11	0	0	0	0	0	0	1957	933	0	0	0	0	0	0	10731	5670	0	0	0	0	0	0
-1	11	11	0	0	0	0	0	0	918	933	0	0	0	0	0	0	5650	5670	0	0	0	0	0	0

An excerpt of the generated dataset is provided in Table IV. The dataset extract has 24 instances of which 12 are from the malicious class and 12 are from the legitimate class. The instances are identified by the classifier as malicious or legitimate by a class label at the beginning of each instance and hence makes the classification independent of IP addresses. A label "1" indicates that instance is malicious while a label "-1" indicates that the instance in legitimate. The complete dataset for the research work has been shared on GitHub [31].

VII. RESULTS AND DISCUSSION

The one-class SVM classifier is first subjected to a training phase, where a trained model is obtained. The model captures the behaviour of the IPs with respect to time based traffic flow features. Before the training phase, the SVM parameters of the classifier are setup for one-class classification. During the training phase, the classifier ideally creates a decision boundary from the data, and any data points appearing outside the decision boundary in the test phase, are considered to be attack instances.

In this classification problem, the effectiveness of the classifier is determined by 3 matrices: sensitivity, specificity and classification accuracy. These matrices are calculated from the true positive (TP), false positive (FP), false negative (FN) and true negative (TN) obtained from a confusion matrix. A confusion matrix [32] (Table V) which is also called a contingency table, shows the number of instances correctly or incorrectly predicted by a classification model.

TABLE V: CONFUSION MATRIX

Actual	**Predicted**	
	Malicious	***Legitimate***
Malicious	True Positive (TP)	False Negative (FN)
Legitimate	False Positive (FP)	True Negative (TN)

The classifier is most effective if it has a high sensitivity value and a high specificity value. The combination of both the sensitivity and the specificity is reflected in the classification accuracy. The TP, TN, FP, TN are first calculated for each of the DoS attacks (Table VI) as each of these attacks are different from one another and may have different detection accuracies.

TABLE VI: CLASSIFICATION OF DOS ATTACKS

Attacks	**TP**	**FN**	**TN**	**FP**
ICMP Flood	15	0	15	0
Ping of Death	63	0	55	8
UDP Flood	34	1	34	1
TCP SYN Flood	52	0	48	4
TCP Land Attack	44	0	42	2
DNS Flood	35	0	34	1
Slowloris	188	250	404	34

1. Sensitivity

The sensitivity is the proportion of true positives in the testing dataset that are correctly identified as such by a classifier (equation 2).

$$Sensitivity = TP / (TP + FN) \qquad (2)$$

The sensitivity of the classifier in classifying the various DoS attacks is given in Table VII.

TABLE VII: SENSITIVITY OF DOS ATTACKS

Attacks	Sensitivity
ICMP Flood	1
Ping of Death	1
UDP Flood	0.97
TCP SYN Flood	1
TCP Land Attack	1
DNS Flood	1
Slowloris	0.43

The sensitivity column in Table VII show the strength of the classifier in being able to correctly classify the malicious (DoS attack) test instances as such. A very high sensitivity for 6 out of the 7 attacks is achieved. The sensitivity for ICMP flood, Ping-of-death, TCP SYN flood, TCP LAND attack and DNS flood is a maximum of 1 and the sensitivity for UDP flood is 0.97. The Slowloris attack however has a low sensitivity value of 0.43 implying a weak detection of the attack by the classifier. The weak detection of the Slowloris attack is due to its attacking nature where malicious HTTP traffic is generated at a very slow rate which is able to exhaust the webserver's HTTP connections without being effectively detected by the classifier.

2. *Specificity*

The specificity is the proportion of true negatives in the testing dataset that are correctly identified as such by a classifier (equation 3).

$$Specificity = TN / (TN + FP) \qquad (3)$$

The specificity of the classifier in classifying the various DoS attacks is given in Table VIII.

TABLE VIII: SPECIFICITY OF DOS ATTACKS

Attacks	Specificity
ICMP Flood	1
Ping of Death	0.87
UDP Flood	0.97
TCP SYN Flood	0.92
TCP Land Attack	0.95
DNS Flood	0.97
Slowloris	0.92

The specificity column in Table VIII shows the strength of the classifier in being able to correctly classify the legitimate (normal traffic) test instances. A high value for specificity is desired for the classification as a high specificity value implies a low false positive rate (false alarm). The classification results show that a high specificity value is achieved for all the DoS attacks. Except for the Ping-of-Death attack with 0.87 specificity score, all attacks achieved well over 90% in specificity measure.

3. *Accuracy*

The accuracy is the overall proportion of the testing dataset that are correctly identified by the classifier. It is calculated using the following equation:

$$Accuracy = TP + TN / (TP + TN + FP + FN) \qquad (4)$$

The overall classification accuracy of the classifier in classifying the various DoS attacks is given in Table IX.

TABLE IX: ACCURACY OF DOS ATTACKS

Attacks	Accuracy
ICMP Flood	1
Ping of Death	0.94
UDP Flood	0.97
TCP SYN Flood	0.96
TCP Land Attack	0.98
DNS Flood	0.99
Slowloris	0.68

The classification accuracy column shows the strength of the classifier in being able to correctly classify both malicious and the legitimate test instances. Except for Slowloris, all other attacks were classified with high accuracy. As discussed earlier the weak detection of the Slowloris test instances can be attributed to a low sensitivity of the classifier to this application-based attack.

VIII. CONCLUSION

In this research we focused on a very critical issue of detecting denial of service attacks from within the cloud, and to do so in a manner, which does not compromise privacy of the cloud users.

The proposed IDS has showed promising results in detecting the 7 types of DoS attacks which are broadly categorized as volume, protocol, and application based attacks.. The novelty of the research work has been the use of time-based traffic flow features with the one-class SVM algorithm in the design of the detection system. Along with the proposed IDS mechanism, this paper provides insights into deploying a test bed on the cloud for experimentation. In addition the dataset generated in this research has been made publically available to encourage further research in this area.

Future work on this research can focus on measuring the target VM's performance under various DoS attacks thus determining the impact of each DoS attack on the target VM.

In addition, the performance of the proposed DoS IDS under normal and DoS attack scenarios can be evaluated. This will show the system resource (CPU, memory) overhead produced by the DoS IDS when it is in execution in the cloud environment.

References

[1] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, “Cloud computing — The business perspective”, Decision Support Systems, vol. 51, 2011, pp. 176-189.

[2] Cisco Systems Inc., “Cisco Global Cloud Index: Forecast and Methodology, 2013–2018”, Cisco Systems Inc. Americas Headquarters, San Jose, CA, USA. 2014.

[3] International Data Corporation, Public Cloud Computing to Reach Nearly $70 billion in 2015 Worldwide, According to IDC, [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS25797415

[4] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and reseach challenges," Journal of Internet Services and Applications, vol. 1, pp. 7-18, 2010.

[5] Cisco Systems Inc., “Cisco 2014 Annual Security Report”, Cisco Systems Inc. Americas Headquarters, San Jose, CA, USA. 2014.

[6] Juniper Network Inc., “White Paper - Defending Against Application-Layer DDoS Attacks”, Juniper Network, Inc., Sunnyvale, CA, USA. Dec 2013.

[7] DDoS Attacks, Incapsula [Online]. Available: http://www.incapsula.com/ddos/ddos-attacks/

[8] Snort, Snort [Online]. Available: https://www.snort.org/

[9] A. M. Lonea, D. E. Popescu, O. Prostean, and H. Tianfield, "Evaluation of Experiments on Detecting Distributed Denial of Service (DDoS) Attacks in Eucalyptus Private Cloud," Soft Computing Applications, Advances in Intelligent Systems and Computing, vol. 195, pp. 367-379, 2013.

[10] C. J. Chung, P. Khathar, T. Xing, J. Lee, and H. Dijiang, "NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems," IEEE Transactions on Dependable and Secure Computing, vol. 10, 2013.

[11] S. Gupta, P. Kumar, and A. Abraham, "A Profile based Network Intrusion Detection and Prevention System for Securing Cloud Enviroment," International Journal of Distributed Sensor Networks, vol. 2013, 2013.

[12] OpenNebula Flexible Enterprise Cloud Made Simple, OpenNebula [Online]. Available: http://opennebula.org/

[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. 42, pp. 42-57, 2012.

[14] V. Engen, J. Vincent, and K. Phalp, "Exploring Discrepancies in Findings Obtained with the KDD Cup 99 Data Set," International Journal of Intelligent Data Analysis, vol. 15, pp. 251-276, 2011.

[15] S. Singh and M. Bansal, "Improvement of Intrusion Detection System in Data Mining using using Neural Network," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, pp. 1124-1130, 2013.

[16] R. Shea and J. Liu, "Performance of Virtual Machines Under Networked Denial of Service Attacks: Experiments and Analysis," IEEE Systems Journal, vol. 7, pp. 335-345, 2013.

[17] B. Loganayagi, S. Sujatha, “Improving Cloud Security through Virtualization”, Communications in Computer and Information Science, vol. 204, S. Barbosa, P. Chen, X. Du, J. Filipe, O. Kara, T. Liu, I. Kotenko, K. Sivalingam, T. Washio, eds., Springer, Heidelberg, 2011, pp 36-45

[18] R. Beverly, S. Bauer. The spoofer project: inferring the extent of source address filtering on the internet. Usenix Struti, 2005.

[19] S. Shiaeles, V. Katos, A. Karakos, B. Papadopoulos, “Read time DDoS detection usinf Fuzzy Estimators”, Computers and Security, vol. 31, p. 782, 2012.

[20] Hewlett-Packard, HP Helion Eucalyptus (Online). Available: http://www8.hp.com/us/en/cloud/helion-eucalyptus-overview.html

[21] Eucalyptus Documentation, Eucalyptus [Online]. Available: http://docs.hpcloud.com/eucalyptus/4.1.2/

[22] Christoper J.C. Burgers, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998, pp. 121–167.

[23] B. Sch¨olkopf, J. Platt, J. Shawe-Taylor, J. Smola, and R Williamson. Estimating the support of a high-dimensional distribution. Neural Computation, vol. 13, 2001, pp. 1443–71

[24] M. Amer, M. Goldstein, S. Abdennadher, Enhancing One-class Support Vector Machines for Unsupervised Anomaly Detection. ACM SIGKDD Workshop on Outlier Detection and Description, ACM New York, USA 2013, pp 8-15

[25] K. Hempstalk, E. Frank, I. Witten, “One-class Classification by Combining Density and Class Probability Estimation”. Machine Learning and Knowledge Discovery in Databases, W. Daelemans, B. Goethals, K. Morik (eds), Springer, Heidelberg, 2008, pp 505-519

[26] L. Manevitz, M. Yousef, “One-Class SVMs for Document Classification”, Journal of Machine Learning Research, vol 2, 2001, pp. 139-154.

[27] Y. Chen, X. Zhou, and T. Huang, “One-Class SVM for Learning in Image Retrieval”, IEEE International Conference on Image Processing, 2001.

[28] C. Chang, and C. Lin, “LibSVM: A library for Support Vector Machines”. ACM Transactions on Intelligent Systems and Technology, vol. 2, 2011, pp. 1-27

[29] P. Wang, W. Huang, C. Varela, “Impact of Virtual Machine Granularity on Cloud Computing Workloads Performance,” 11th IEEE/ACM International Conference on Grid Computing, 2010, Brussels, pp. 393-400.

[30] S. Sanfilippo, Hping (Online). Available at http://www.hping.org/

[31] R. Kumar, S. Lal, A. Sharma, GitHub [Online]. Available: https://github.com/uspscims/cloudsecurity

[32] P. Srinivasulu, D. Nagaraju, P. Kumar, and K. Rao, “Classifying the Network Intrusion Attacks using Data Mining Classification Methods and their Performance Comparison”, International Journal of Computer Science and Network Security, vol. 9, 2009. pp. 11-18.