Virtual Host based Intrusion Detection System for Cloud

Manthira Moorthy S [#1], Rajeswari M [#2]

[#]Department of Computer Science and Engineering, Hindustan University
P.O.Box No.1, Rajiv Gandhi Salai (OMR),
Padur, (Via) Kelambakkam,
Chennai - 603 103, India .
[1] moorthysgm@gmail.com
[2] rajeswarim@hindustanuniv.ac.in

***Abstract*—Cloud computing is an internet or intranet based computing, where infrastructure, information and Application are provisioned based on demand. The major breach in cloud is its security due to its huge extends of resources available in it. The major threats are data loss or leakage and hijacking. This paper presents Cloud Intrusion Detection Data Sets (CIDD) and virtual host based Intrusion Detection System. CIDD contains attack signatures based on port that are opened in cloud for communications, The signatures are prepared manually and scored using common vulnerability scoring system. Genetic Algorithm is the technique applied for generating rules from existing datasets. Large set of rules can be generated for intrusion detection by mean of genetic operation.**

Keyword- Cloud computing, Cloud security, Intruder, Cloud Intrusion Detection Datasets, Genetic Algorithm.

I. INTRODUCTION

Cloud Computing is the latest trend in computing. Cloud Computing provides computing resources that are delivered as a service over internet. Cloud consists of hardware and software resources made available on the internet. Anyone can easily provide, manage and sustain cloud resources for a fraction of cost. Due to enormous amount of storage, backup and restore facility more number of people and organisations have moved to cloud. So there is a critical need for secure data storage and secure access in the cloud. The Test-bed environment used in this paper is Cloud Stack with KVM hypervisor. Cloud Stack is an open source cloud computing tool for creating, managing, and deploying cloud infrastructure and services. It uses existing hypervisors such as KVM, vSPhere and Xenserver for virtualization. In Cloud based intrusion detection, anomaly-based approaches in particular suffer from accurate evaluation, comparison, and deployment which originates from the scarcity of adequate datasets. Many such datasets are heavily anonymized and do not reflect Cloud, These deficiencies are primarily the reasons why a perfect dataset is yet to exist for cloud. Here we proposes Cloud intrusion datasets which are prepared base on ports that are opened in cloud stack that are opened in cloud for communication. The Ports opened in Cloud Stack Components for communication are given in TABLE I. Cloud stack installation consists of two machines, one machine running the Management Server and another machine running KVM hypervisor. The traffic coming into the cloud stack should be monitored by means of IDS for malicious activities or policy violation. As traffic flow to hypervisor only through management server both management server and KVM hypervisor are considered as single host virtually. IDS is developed to monitor the traffic to entire host. Hence it is called as Virtual Host based Intrusion Detection.

TABLE I
Ports used by Cloud Stack Components

Components	Type	Port	Details
Cloud Management Server	TCP	9090	To/from Cloud Management Server
User /Client /API	TCP	8080	User /Client /API to Cloud Management port (Authenticated Mode)
User /client	TCP	8096	User /Client /API to Cloud Management port (Un-Authenticated Mode)
MySQL	TCP	3306	Cloud Management Server to My SQL
KVM	TCP	22	Cloud Management Server to KVM
NFS	TCP	111	Cloud Management Server to NFS

II. RELATED WORK

IDS can be designed using different kind of soft computing techniques as below

A. Artificial Neural Network based IDS

Artificial Neural Network (ANN) in intrusion detection is used to generalize data from incomplete data and able to classify data as being normal or intrusive. Kleber Vieira et al. as in [7], in their paper introduce Intrusion detection model for cloud. In this architecture each node of the cloud contains IDS which provides interaction among service offered. The limitation of this approach is that it cannot detect any insiders in the VM's and requires more training samples as well as more time for detecting intrusions effectively.

B. Support Vector Machine based IDS

SVM is used to detect intrusions based on limited sample data, where dimensions of data will not affect the accuracy. H. Lei et al.as in [4] , designed an intelligent module for network intrusion prevention system with a combination of SNORT and configurable firewall. The support vector machine (SVM) classifier is also used with SNORT to reduce false alarm rate and improve accuracy of IDS.

C. Genetic Algorithm based IDS

Genetic algorithms (GAs) are used to select network features or to determine optimal parameters which can be used in other techniques for achieving result optimization and improving accuracy of IDS. Todd Vollmer et al. as in [11], in their paper introduce a multi-modal genetic algorithm solution for autonomous rule creation. This algorithm focuses on the process of creating rules once an intrusion has been identified, rather than the evolution of rules to provide a solution for intrusion detection. Output rules were sorted according to a fitness value and any duplicates were removed.

D. Types of IDS for Cloud

There cloud are four types of IDS used in cloud they are,

- Host based IDS,
- Network based IDS,
- Hypervisor based IDS,
- Distributed IDS.

Host based IDS is used to monitor traffic to a specific host. Y.Guan et al.as in [13], proposed using change point based idea to detect all types of attacks. This approach is based on statistics and probability theory. In this approach, all attacks are taken as a sample space. Then the set is decomposed using statistics based on mutually exclusive sets. The generated subsets which belong to sample space are used to construct intrusion detection algorithm. However, no experimental results or deployment issues are reported yet. Distributed IDS (DIDS) consists of several IDS over a large network, all of which communicate with each other, or with a central server that enables network monitoring. Hisham A. Kholidy et al.as in [5], proposed a framework for cloud-based IDS. A distributed architecture without central element is proposed, balancing the workload across the nodes of the cloud and thus avoiding a single point of failure for not having central element. However, the constant exchange of information between nodes to maintain the consistency of the databases, can reduce system performance.

Network based IDS detect malicious activity by monitoring network traffic. A.Bakshi et al.as in [1], proposed an architecture for detecting DDoS attack in VM. IDS systems are installed in virtual switch to log coming or outgoing traffic into database. To detect known attacks, the logged packets are analyzed and compared by the IDS in real time with known signature. This approach can block the DDoS attack in virtualized environment and can secure services running on virtual machines. But it cannot detect all types of attacks as the tool used here is snort. It identifies only known attacks. C.Mazzariello et al.as in [2], proposed snort based misuse detection in open source eucalyptus Cloud. In this approach, snort is deployed at Cloud controller (CC) as well as on physical machines (hosting virtual machines) to detect intrusions coming from external network. This approach solves the problem of deploying multiple instances of IDS. It is a fast and cost effective solution. However, only known attacks are detected as snort is involved.

E. Cloud Intrusion detection datasets

Hisham A. Kholidy et al.as in [6], proposed a Cloud Intrusion Detection Dataset (CIDD) that is the first one for cloud systems and that consists of both knowledge and behavior based audit data collected from both UNIX and Windows users. However the datasets are not sufficient for intrusion detection in cloud.

F. Limitations in the Existing System

Self-learning mechanism cannot be efficient for cloud as it require quick detection. The training time is high so it cannot be used for real time intrusion detection. SVM is effective only for low sample of data. In host based intrusion detection are host specific and not able to monitor entire network where in network based IDS are able to detect only known attacks. The continuous exchange of information between the nodes in distributed IDS reduces the performance of the system. There is no effective cloud intrusion detection datasets to correlate the attacks in cloud.

III. SECURITY ARCHITECTURE FOR CLOUD

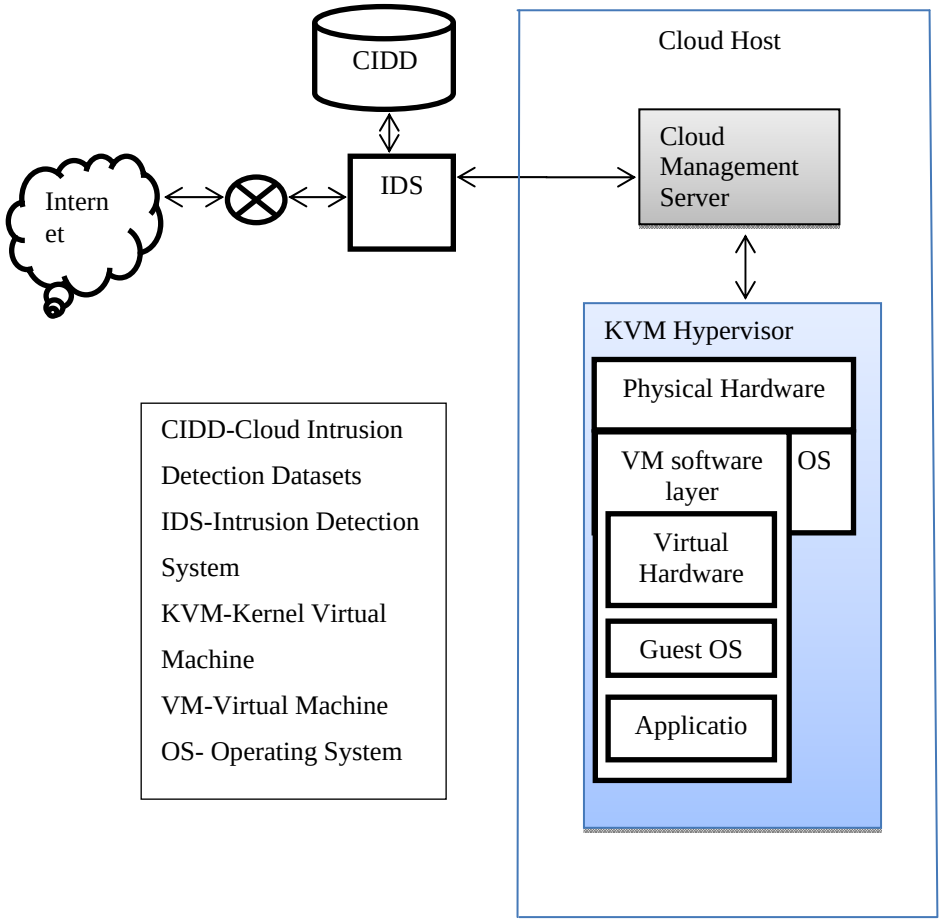

Fig. 1. Security architecture for cloud

Ideal network architecture is to filter the network traffic before it enters inside the cloud infrastructure. To do so the Host based intrusion detection system is placed between router and Cloud Host in Security architecture for cloud is represented in Fig. 1. The router is used to filter the traffic by using IP access control list known as ACL. The configuration of ACLs allows filtering network packets based on their source, destination, protocol and port. ACL cannot examine the contents carried out by the Packets. The IDS is placed below the router in order to examine the contents carried out by the packet. IDS examine the content against the cloud intrusion detection datasets Signatures. If IDS correlate any attacks it drops the request and alerts the user.

IV. DESIGN OF IDS FOR CLOUD

A. Cloud Intrusion Detection Datasets

Cloud intrusion detection datasets is a systematic approach to generate the required dataset. This is based on the concept of profiles which contain detailed descriptions of intrusions, protocols, or lower level network entities for cloud. Real traces are analyzed to create profiles for cloud. Various multi-stage attacks scenarios were subsequently carried out to supply the anomalous portion of the cloud intrusion detection datasets. The vulnerabilities are scored by means of Common Vulnerability Scoring System (CVSS) [9]. CVSS consists of three groups:

- Base
- Temporal
- Environmental

In this Paper Base Metric is considered as other metrics are optional. The base metric group captures the characteristics of vulnerability that are constant with time and across user environments.

B. Genetic Algorithm

A Genetic Algorithm (GA) is a optimization technique for generating new rules in cloud Intrusion detection system. The evolution usually starts from a population of randomly generated individuals. Here the individuals

are Intrusion detection rules. In each generation, the fitness of every rules in the population is evaluated, multiple rules are selected from the current population based on their fitness, and modified by recombination and mutation to form a new rules. The new rules is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the rules [12].

Algorithm:

Step 1 : Initialization: Generation of Initial population with Fuzzy if – then rules

Step 2: Evaluation Test: Perform evaluation test to calculate goodness of newly generated Rules.

Step 3: Generation : Generation of new Rules using Genetic operation

Step 4 : Replacement : Replace a part of current population with newly generated rules and return to step 2 until maximum iteration

Step 5: saving the best individual of the algorithm. Terminate the Algorithm if satisfied or goto step 1

Step 6: Terminate the algorithm if maximum iteration is achieved.

1) Initial Population:

Initial populations are the malicious traffic collected from network by means of network sniffers. Sniffers are used to record network traffic without doing harmful.

2) Fitness Function:

Fitness function is depends up on total number of attacks and the number of True positive and false negative. True Positive is the detection of correct or true request as attack. False negative is the failure to detect the real attack.

Fitness Function is given by,

$$F(A_i) = \frac{\alpha}{A} + \frac{\beta}{B} \quad (1)$$

A - Total Number of attacks

B - Total Number of connections

α - Number of true Positive's

β - Number of False positive's

From equation 1, we can calculate the fitness score for an individual. Fitness function is used for the selection of best individual for the generation of new rules by cross over and mutation.

TABLE II
Intial Population

	Port	Protocol	Content	Attack-Type
A1	8080	TCP	P2P-Dest	attempted-user
A2	8080	TCP	aim|3A|goaway?message=	misc-attack
A3	8080	TCP	/forum/links/public_version.php	trojan-activity
A4	443	TCP	|7C|vv|7C|	trojan-activity
A5	8080	TCP	JOIN|20 23 21|xd|21 20|r0x	trojan-activity

TABLE III
Fitness Table

	True Positive	False Negative	Fitness Value
A1	4	0	0.8
A2	4	1	0.9
A3	3	1	0.7
A4	2	3	**0.7**
A5	2	3	**0.7**

3) Cross over and mutation

The Best-fit individuals undergoes cross over and mutation for generating new rules. This newly generated rules, is used to classify the attacks that is not in the training data sets.

For Example,

parent1 - Protocol TCP, Port Number 8080, Content "JOIN|20 23 21|xd|21 20|r0x"

parent2 - Protocol TCP, Port Number 443, Content "|7C|vv|7C|"

After Single crossover, the newly generated rules are

offspring 1- **Protocol TCP**, **Port Number 8080,** content "|7C|vv|7C|",

offspring 2 - Protocol TCP, Port Number 443, content "**JOIN|20 23 21|xd|21 20|r0x"**

After Mutation,

offspring 1- Protocol UDP, **Port Number 8080,** content "|7C|vv|7C|"

offspring 2 - Protocol UDP, Port Number 443, content "**JOIN|20 23 21|xd|21 20|r0x"**

C. IDS for cloud

Intrusion detection system (IDS) is used to monitors cloud network or system activities for malicious activities or policy violations and performs active or passive measures. It consist of three components namely,

- Event Auditor
- IDS service
- CIDD

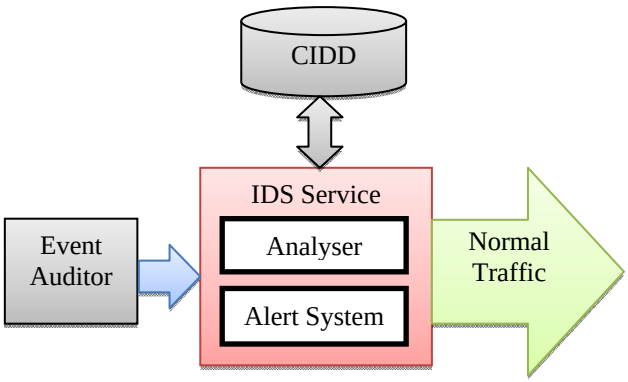

Fig. 2. Functional Block Diagram of IDS

1) Event Auditor

Event Auditor captures the network traffic to cloud management server. It decodes the packet. Information contained in packet such as Source address, Destination address, Port numbers, Protocol are derived from the header portion and content is derived from Payload portion.

2) IDS service

The (IDS) service increases cloud security level by knowledge-based method that detects known trails left by attacks or certain sequences of actions from a user who might represent an attack. The audited data is sent to the IDS service core, which analyzes the data with predefined rules. This has two subsystems namely analyzer system and Redirection Engine.

- *Analyzer system*

The analyzer receives audit packages and it matches with the rules in the storage service. The matching is done by means of pattern matching. If matching criteria is found it response the result to IDS Service Core.

- *Alert System*

This subsystem will work when intrusion is detected. The IDS service will discard the packet and alert the user.

3) CIDD

The CIDD is a database system which contains knowledge based service. Knowledge service is based on set of predefined rules for known attacks. Whenever a Management Server gets requests or responses, the analyzer system compares the information in the cloud intrusion detection datasets.

V. Performance Analysis

A. CIDD Effectiveness

Cloud Intrusion Detection Datasets Effectiveness can be determined by calculating True Positive Ratio and False Positive Ratio as below.

1) True Positive Ratio

True Positive ratio is used to evaluate effectiveness of Cloud Intrusion datasets. Effectiveness is determined by sending attacks towards the cloud IDS in order to detect them. Attack has been send from attackers to the Management server. To evaluate the number of attacks detected by Cloud IDS. The whole injection composed of 50 random attacks. 40 out of 50 injections are alerted. This mean that 10 attacks are not found by cloud IDS. True Positive Ratio is calculated using the following formula

$$TPR = \frac{TP}{TP + FN} \times 100$$

Where,

TP – An Attack has occurred and alarm has been raised

FN- An Attack has occurred and but no alarm was raised

$$TPR = \frac{40}{40 + 10} \times 100$$

TPR = 80 %

The acceptable level of true positive ratio is 60%as in [10].In cloud IDS the TPR is 80% which makes 20% above the acceptable level is achieved. Based on the fact the Cloud intrusion detection datasets are 80% effective.

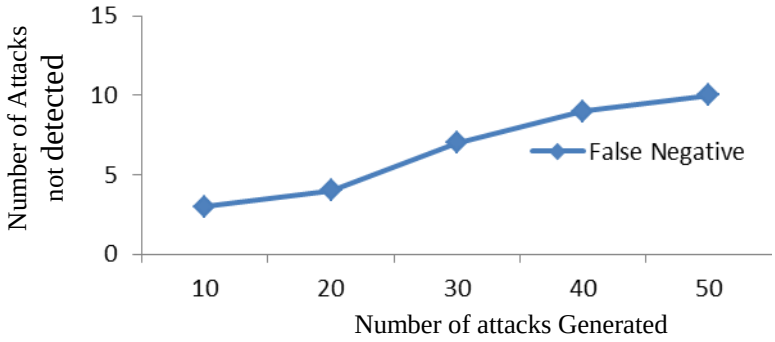

Fig. 3: False Negative

2) False Positive Ratio

False Positive Ratio is used to determine the proportion false alert. The darpa Datasets along with 50 attacks, which result in a total of 400 packets are send to Management Server. Over this large traffic, 40 alerts are generated by cloud IDS. This result is same to the True Positive Ratio experiment. This conclude that injection of back false positive alerts ground traffic did not raise any false positive alert. So False Positive Ratio with FP=0 is calculated using the following formula

$$FPR = \frac{FP}{FP + TN} \times 100$$

Where , TN = (Total packets – Attack Packets)

= 400-50 =350

$$FPR = \frac{0}{0 + 350} \times 100$$

= 0%

The FPR for cloud IDS is 0%. An Ideal Intrusion detection System should have False Positive Ratio of 0%. So Cloud IDS is ideal

B. Efficiency of IDS

Latency is the measure of Time delay to detect malicious packets with the network traffic. To do so, 50 attacks are sent with increasing background traffic and time taken to detect 50 attacks is determined. The background traffic is increases by 0.5 Mbps steps.

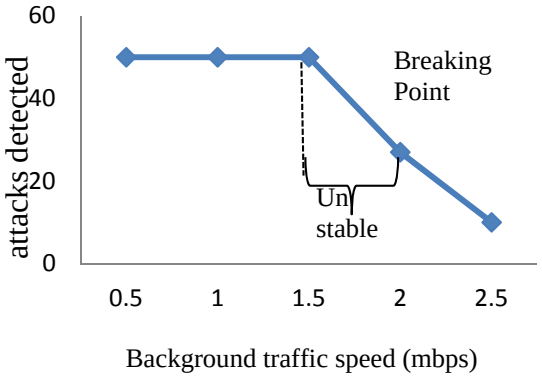

Fig. 4 : Latency in Intrusion Detection

Fig. 4., show that when background traffic increases the time taken for filtering packet also increases. The increase in latency shows that the cloud IDS queue the network packets destined to the Cloud Management Server. During high load traffic cloud IDS struggle to reply to each packet. However, IDS never drops a packet and always detect correct number attacks. When background traffic increases 2 mbps cloud IDS generates an error and stops itself. This error is called "segment fault". This is due to insufficient RAM. Therefore an unstable interval was determined between 1.5 and 2 mbps.

VI. CONCLUSION

Cloud intrusion detection datasets are able to detect cloud attacks. Cloud based IDS were able to detect 80% of Random sets of cloud attacks. By adding background traffic retrieved from darpa, IDS was able to detect the same percentage of attacks and no false positive alarm is raised while filtering background traffic. The efficiency of cloud IDS is determined by injecting attacks with increasing background traffic. Result show that latency is increasing according to background traffic. This does not have effect on cloud intrusion datasets. However, a breaking point was identified at 2 mbps, the cloud IDS generated an error and stopped. Therefore, an unstable interval was determined between 1.5 to 2 mbps.

VII. FUTURE ENHANCEMENT

The work presented in the paper has fulfilled some gaps. Further work, in this area could be carried out in order to make universal cloud intrusion datasets. The efficiency of cloud based IDS can be improved by implementing multiple IDS over the cloud and installing Multiple Management servers.

REFERENCES

[1] A.bakshi, and B. Yogesh. Securing Cloud from DDOS Attacks Using Intrusion Detection System in Virtual Machine, *in proc. Second International Conference on Communication Software and Networks*, 2010, pp. 260-264.

[2] C. Mazzariello, R. Bifulco, and R. Canonoco. Integrating a network IDS into an Open source Cloud computing, *in proc. Sixth International conference on Information Assurance and Security (IAS)*, 2010, pp. 265-270.

[3] David.J.Chaboya, Richard.A.Raines, Rusty.O.Baldwin, Barry.E.Mullins,"Network Intrusion Detection:Automated and Manual method prone to Attack and Evaton",*IEEE Security and Privacy* ,vol. 4,Iss:6, page no.36-43,2006.

[4] H. Li, and D. Liu. Research on Intelligent Intrusion Prevention System Based on Snort, *in proc. International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE)*, vol. 1, pp. 251-253,2010.

[5] Hisham A. Kholidy, Fabrizio Baiardi. CIDS: A framework for Intrusion Detection in Cloud Systems, *in proc. Ninth International Conference on Information Technology- New Generations*, 2012, pp. 379-385.

[6] Hisham A. Kholidy, Fabrizio Baiardi. A cloud intrusion detection dataset for cloud computing and masquerade attacks, new generations, *in proc. Ninth International Conference on Information Technology- New Generations*, 2012, pp. 397-402.

[7] Keleber Vieira, Alexandre Schulter, Carlos Becker Westphall, And Carla Merkle Westphall. Intrusion Detection For Grid And Cloud Computing, *Ieee Press Cybersecurity*, 2010.

[8] Omid Mahdi , Harleen kaur "An Efficient Hybrid Honeypot Framework for Improving Network Security" International Journal of Computer Science and Information Security, Vol. 9, No. 2, 2011

[9] Peter Mell, Karen Scarfone, Sasha Romanosky, A Complete Guide to the Common Vulnerability Scoring System Version 2.0, 2007

[10] (2010), Symantec [online], Available : http://www.symantec.com/connect/articles/strategies-reduce-false-positives-false-negatives-nids.

[11] Todd vollmer, Jim Alves-Foss, Milos Manic, Autonomous Rule Creation for Intrusion Detection, *IEEE symposium on computational Intelligence in Cyber Security*, 2011.

[12] Wei Lei, " Using Genetic Algorithm for Network Intrusion Detection," *Mississipi State University*, 2004.

[13] Y. Guan, and J. Bao. A CP Intrusion Detection Strategy on Cloud Computing, *International Symposium on Web Information Systems and Applications (WISA)*, 2009, pp. 84–87.