IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-03, Issue-06 E-ISSN: 2347-2693

HTTP service based Network Intrusion Detection System in Cloud Computing

Sudhansu Ranjan Lenka [1*] and Bikram Keshari Rath [2]

[1*] *Computer Science & Engg, Trident Academy of Technology, India*
[2] *Computer Science & Application, Utkal University, India*
sudhansulenka2000@gmail.com

www.ijcseonline.org

Received: Jun /08/2015 Revised: Jun/13/2015 Accepted: Jun/18/2015 Published: Jun/30/ 2015

Abstract— Recently, the usages of Cloud Computing are increasing rapidly and gained tremendous success over the internet. Therefore, security is the major challenge in Cloud computing and one of the major issues is to protect the Cloud resources and the services against network intrusions. So Network Intrusion Detection System (NIDS) are installed in the Cloud networks to detect the intrusions in the system. In this paper we proposed an NIDS based on Naïve Bayes Classifier to be implemented in Cloud. The main aim of the NIDS is to improve the performance by preparing the training dataset which can detect the malicious connections that exploit the Cloud HTTP services. In the training phase, the Naïve Bayes Classifiers select the important Network traffic that can be used to detect the attacks. In the testing and execution phases the proposed IDS using the Naïve Bayes Classifier classifies the services based on the selected features into normal or attacks. The proposed IDS carried out on NSL-KDD'99 dataset and results in high detection with low false alarm as compared with other similar IDS.

Keywords—*Cloud Computing; Cloud security;Network based Intrusion Detection System; Naïve Bayes Classifier*

I. INTRODUCTION

These days, most of the services are processed through a computer networks over the internet. The tremendous rises in the computer networks and huge accessibility of internet have gained a lot of positive aspects. However, the attackers may access the confidential information or network resources by exploiting the internet. To avoid anomaly intrusions in the Cloud network some security tools are used such as firewalls, antivirus software and Intrusion Detection System (IDS). The traditional attacks such as IP spooling, Address Resolution Protocol (ARP), port scanning, Denial of Service (DoS), Distributed Denial of Service (DDoS) etc can be detected using firewall. But the firewall can only detect the network intrusions at the boundary of the network, inside attacks cannot be captured by it. Therefore, to handle all types of intrusions firewall is not an efficient technique.

NIDS is another technique to detect the intrusions in Cloud Computing. It acts as an alert system and raise alarm when any unauthorized user wants to penetrate through the system. The accuracy of the NIDS depends on three parameters, the first parameter is the detection technique (signature based or anomaly based), the second parameter is the installation location (front end or back end) and last parameter is the configuration technique (distributed or centralized).

In this paper, we propose a NIDS based on Naïve Bayes Classifier to detect the intrusions and also the positioning of NIDS in the Cloud networks has been recommended. Bayesian classifier implements the statistical method to classify the network events as normal or as attacks. In the training phase, the algorithm classifies the dataset into normal and attack classes by computing the conditional probabilities for each of the different classes. In the testing and execution phase, classifier classifies the unknown network traffic based on the probability value of different classes; therefore, the unknown network request can be classified as the class that generates maximum probability value [1, 2]. Our main objective is to decrease the impact of intrusions and ensuring higher detection rate with reduce false alarm rate.

Rest of the paper is organized as follows: the section II presents the related work and theoretical background, section III presents the detailed description of the proposed NIDS, implementation and results of the proposed NIDS is given in section IV and finally conclusion and future works are described in section V with the references at the end

II. RELATED WORK AND THEORETICAL BACKGROUND

A. *Problem Statement*

The objective is to design an efficient NIDS that can detect the intrusions in Cloud network with reduced false positives and false negatives.

Corresponding Author: *Sudhansu Ranjan Lenka,*
,sudhansulenka2000@gmail.com
Department of Computer Science., Trident Academy of Technology., India

B. *Related Work*

In [3] the IDS is proposed based on anomaly detection approach, it implements the combination of K-means, K-Nearest Neighbor classifier and Naïve Bayes Classifier. It uses entropy based algorithm for feature selection and classifies the attacks into different categories like DOS, U2R, R2L, and probe. The performance of the IDS is good but the computational time may increase.

In [4] a score based multi-cycle detection algorithm is proposed based on Shiryaev-Roberts procedure. This procedure is computationally inexpensive and it can be easily implemented in real time IDS. The proposed IDS minimize the detection rate, however, it increases the false alarm rates. Therefore, it implements an additional filtering technique to increase the detection accuracy, which it may increase the processing delay.

A tree based IDS classification algorithms are proposed in [5]. Implementing the Random Tree model it could achieve 97.47% of detection accuracy and its false rate is about 2.5%.

In [6] the IDS based on the combination of Snort and Bayes theorem has been proposed to be implemented in Cloud Computing. In this, Snort checks the captured packets using signature based detection and Bayesian Classifier identifies the intrusion packets and normal packets. The IDS achieves the detection rate of about 96% and 1.5% false positive rate.

In [7] a virtual machine compatible IDS have been proposed that consists of two main components: IDS management unit and IDS sensor. IDS management unit includes event gatherer, database management system, analysis component and remote controller. IDS sensor capture the malicious events and transmits the events to event gatherer and stores it in event database. Analysis component analyze the stored events as per design. IDS-VMs are controlled by the IDS Remote Controller which can communicate with other IDS-VMs and IDS sensors. Here sensor can be treated as a NIDS and it is configured by IDS remote controller. The function of IDS-VM is to controls, monitors and configures the VM. This technique prevents the VMs from being compromised but requires multiple instances of IDS.

C. *Theoretical Backgroundt*

1) Bayesian Classifier:

It operates on a strong independence assumption [8]. This means the effect of an attribute value on a given class does not depend on other attributes. The classifier follows the statistical approach which can predict the probability of a network request belongs to a normal or attack class.

Let X is a data tuple. H is the hypothesis that X belongs to a particular class C. P(H|X) is the posterior probability of H conditioned on X. Using Bayesian theorem, the probability P(H|X) of a hypothesis H on a given data tuple X can be defined by Eq. (1)

$$P(H \mid X) = \frac{P(X \mid H)\, P(H)}{P(X)} \qquad (1)$$

Let D is a training set of tuples and their related class labels. Each tuple is represented by a vector X=(x1,x2,..xn)and each tuple contains n attributes A1,A2,…An. Assume there are m number of classes C1, C2, …Cm. Classifier will predict that tuple X belogs to class Ci, iff

$$P(Ci \mid X) > P(Cj \mid X) \text{ for } 1 \leq j \leq m \text{ , } j \neq i$$

The maximal P(Ci |X) can be derived from Eq.(2). Since P(X) is constant for all classes, we need to maximize p(X|Ci) P(Ci) which is defined in Eq.(3)

$$P(Ci \mid X) = \frac{P(X \mid Ci)\, P(Ci)}{P(X)} \qquad (2)$$

$$P(Ci \mid X) = P(X \mid Ci)\, P(Ci) \qquad (3)$$

Thus, Bayesian classifier predicts the request packet is normal or attack based on the previously stored packets.

2) *Snort:*

It is an open source IDS implements signature based intrusion detection technique. It is widely used, easily configurable and can run on multiple platforms like GNU/Linux, Windows, etc. It captures the network data packets and compares their contents with the predefined known attack patterns for any correlation.

3. Information gain for feature selection:

Information Gain is used to rank the features individually based on the class labels. Let S be a training data set consisting of m classes. Suppose the data set contains S_i samples belongs to class I, then the information needed to classify the given sample is defined in Eq. (4)

$$\text{Info}(S_1, S_2, .. S_m) = -\sum_{i=1}^{m} \frac{Si}{S} \log(S_i / S) \qquad (4)$$

Suppose feature F with values $\{f_1, f_{2,..}\ f_v\}$ divide the data set into subsets $\{S_1, S_2,.. S_v\}$ and let S_j contains S_{ij} samples of class i. Entropy of features F is defined in Eq. (5)

$$E(F) = \sum_{i=1}^{m} \frac{S1j + S2j + \ldots + Sij}{S} \text{Info}(S_{1j}, S_{2j},..S_i \qquad (5)$$

Information Gain for feature F can be computed using Eq. (6)

$$\text{Gain}(F) = \text{Info}(S_1, S_2, .. S_m) - E(F) \qquad (6)$$

III. PROPOSED NIDS BASED ON BAYESIAN CLASSIFIER IN CLOUD COMPUTING

A. *Integration of NIDS in Cloud*

As shown in fig.1 Cloud Computing consist of two ends i.e. Front end and back end. The end users can access the Cloud offered services through front end.

The front end is connected between the external network and internal network. Processing server executes the users request and allows accessing the VM instances. Internal networks give the design structure of the VM instances. For E.g., each VM has two network IPs named public IP and private IP [9]. Each VM can communicate with each other through private network. The mapping of public IP of VM to private IP of VM can be done through Network Address Translation (NAT). As shown in fig. 1 NIDS module can be installed at different position in Cloud. When the module is positioned at the front end of Cloud then it can only detect the intrusions at the external network but unable to detect internal intrusions. When the module is positioned on the processing server then it can able to detect the intrusions both at the internal as well as external network of Cloud.

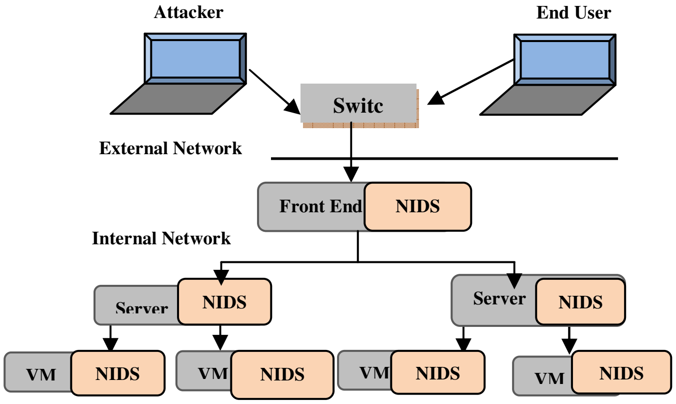

Figure 1. NIDS in Cloud Computing

But the efficiency of the NIDS may be deceases since large number of packets passes through the server. When the NIDS is integrated on each VM then it can detect the intrusion on his/her VM. Such configuration needed multiple instances of NIDS, which makes very difficult to manage the NIDS since the VMs are dynamically migrated, provisioned or de-provisioned

B. *Proposed Framework of NIDS*

As shown in Fig. 2, NIDS module consisting of three main components viz; packet preprocessing, analyzer and storage system.

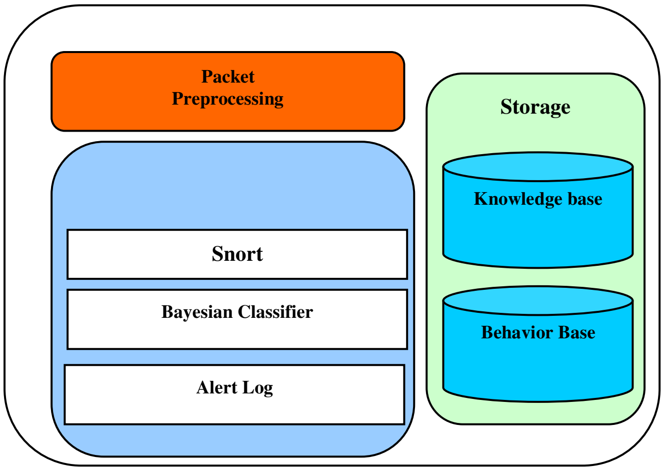

Figure 2 Bayesian Based NIDS

Preprocessing module convert the captured packets to specific format by removing redundant information, that has got very less impact with detection. Analyzer decides whether the captured data packet is normal or intrusion by using signature based and anomaly detection techniques. The analyzer consists of three components: Snort, Bayesian Classifier and Alert Log. Alert Log system logs the intrusion packets and sends alert message to other NIDS. Other NIDS stores these alert messages in their storage device. There are two types of storage devices: Knowledge base and Behavior base. Knowledge base stores the rules or the predefined known attack patterns and this device is used by Snort for intrusion detection. Behavior base stores both the normal and intrusion packets and it is used by Bayesian Classier to detect the intrusions.

NIDS module implements two types of detection techniques: Signature based detection and Anomaly based detection. Signature based detection compares the captured data packets with the rules stored in Knowledge base to detect the known attacks easily. It logs the intrusion packets in alert database, so that other NIDS can communicate the message with each other. Non-intrusion packets are processed for anomaly detection and it is used by Bayesian Classifier to predict the class label (normal or intrusion) by comparing it with behavior base.

C. *Working principle of the proposed NIDS*

In the first phase (Fig. 3), the NIDS read the NSL-KDD training data set [14]. The records are classified into normal or intrusion classes by applying Naïve Bayes Eq.(3). The training data set are reduced by removing the records that generates false alarm and the NIDS is retrained on the reduced data set and the performance is again measured. These reduction processes are repeated until the performance reaches 100%. The aim of the training phase is to identify the records from the training data set that provides 100% accuracy.

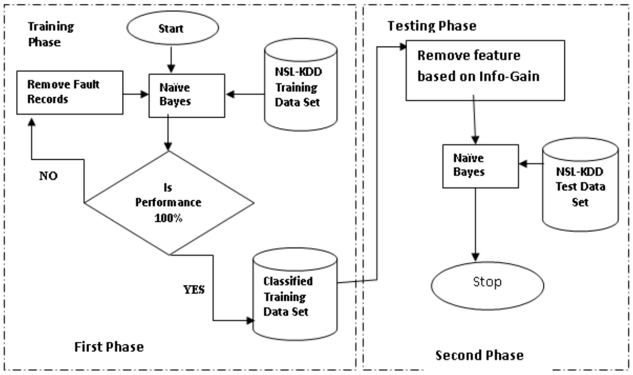

Figure 3. Working Principle of the proposed NIDS

The main aim of the proposed NIDS is to identify HTTP data packets only. So in the second phase, HTTP records are selected from the training data set by removing features 2, 3 and 4 from the 41 features of the NSL- KDD data set. These features include protocol, service and flag. These features are removed because all the HTTP records have identical protocol (TCP), service (HTTP) and flag (SF). In the proposed algorithm, the features are selected based on Information Gain. In this process features are removed from both training and testing data set and then the performance of the classifier is measured using Naïve Bayes Classifier.

D. Proposed Algorithm

Input:

F = All the 41 features of NSL-KDD data set
N = Total number of records in the data set.
CA = classifier accuracy on entire data set.
Err = RMSE on entire data set.
Avg_TPR = average TPR on entire data set.
TH=Threshold value

First Phase

```
1. For each record Ri
       Do
          If Ri generates false alarm, then
          N = N – {Ri} // N contains the resultant   //training data set
           End
```

Second Phase

```
2. S={F}- {2,3,4}     // Remove feature 2, 3 and 4 to identify only HTTP records
       For each feature Fi,
           i.      $INF_i$ = compute information gain
           ii.     If $INF_i$ < TH
           iii.    S=S - {Fi}
 3. Call Naïve Bayes Classifier based on the selected features
```

IV. IMPLEMENTATION AND RESULT

The main aim of the proposed NIDS is to achieve 100% detection rate during the training phase. It means the training data must detect all the intrusion before implementing the testing phase. In the first phase of the NIDS, execute the algorithm on the training data as training as training as well as testing purpose. These steps are repeated until the accuracy reaches 100% i.e. no false alarm is generated.

In the first phase, KDD-Train 20% (25,192 records) has been taken as training well as testing data set. To implement Naïve Bayes Classifier all the 41 features must have numeric value. Table-1 shows the results of each step during execution of the first phase of the proposed algorithm. In step-0 the false-negative alarm is 0.52% and the false-positive alarm is 2.66%. The records causing these false alarms are removed from the training data set. After step-0, the data set reduced to 24,810 records and the test is repeated again. This process is repeated and in step-5 the TP rate and TN rate reaches 100%. After step-5, the data set reduced to 24,262 records and the resultant data set is treated as training data set.

Step	# of Attack Records	# of Normal Records	Total	True positive (%)	False negative(%)	True negative (%)	False positive (%)
0	11431	13379	24810	97.34	2.66	99.48	0.52
1	11199	13379	24578	97.97	2.03	100	0.00
2	10,934	13379	24313	97.63	2.37	100	0.00
3	10889	13379	24268	99.58	0.42	100	0.00
4	10883	13379	24262	99.94	0.06	100	0.00
5	10883	13379	24262	100.00	0.00	100	0.00

Table-1. The experimental results of the first phase of the algorithm

The second phase of the NIDS is to identify the features that can be used to detect the HTTP attacks. In this phase, features 2, 3 and 4 are removed from the training data set (phase-1 records). Features are selected having Information gain more than the threshold value.

Table -2 shows the selected features based on different threshold value. Table -3 shows the results of the second phase of the proposed algorithm. The performance of the algorithm is tested using NSL-KDD Test 21% (11,850 records) data set.

Threshold Value	# of features selected	Selected features
0.01	25	1,2,3,5,7,9,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38
0.05	23	1,2,3,9,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38
0.1	19	2,3,9,20,21,22,23,26,27,28,29,30,31,32,33,34,35,36,38

Table 2. Selected Features with higher Information Gain

Step	No. of Features	Threshold	True positive (%)	False negative (%)	True negative (%)	False positive (%)
0	38	0.00	97.76	2.24	98.993	1.007
1	25	0.01	99.54	0.46	99.49	0.51
2	23	0.05	99.7	0.30	99.84	0.16
3	19	0.1	99.75	0.25	99.46	0.54

Table-3 Experimental results on the selected Features

IDS Index	Intrusion Detection system	Detection Rate %
IDS I	Anomaly based IDS [10]	68.7
IDS II	IDS based on integration of Snort and bayes theorem [6]	96
IDS III	Tree based IDS classification algorithm [5]	97.49
Proposed IDS	HTTP based NIDS using Naïve Bayes Classifier	99.75

Table 4. Comparison of IDS detection rates

The Proposed HTTP based NIDS achieved good detection rate as compared with similar IDS.Table-4 shows the detection rate of the proposed NIDS along with some similar IDS [13].

V. CONCLUSION AND FUTURE WORK

To detect and handle all types of malicious activities in Cloud network, firewall is not an efficient solution. In this paper, we design the framework of NIDS integrating Naïve Bayes Classifier and Snort to identify the intrusions in Cloud Computing. Additionally, an algorithm has been proposed to classify the data set based on HTTP services. The objective of the NIDS is to enhance the performance by selecting only the important features that identify each attack and normal connections of HTTP services. The proposed NIDS shows significant performance, the TP = 99.75%**,** TN=99.46%, FP=0.54% and FN=0.25%. The performance is better as compared to similar IDS.

As a future work, the proposed NIDS can be installed in Cloud to protect network against real time intrusions. Additionally, it can be applied to other network services like TELNET and FTP.

REFERENCES

[1] Dewan F, Mohammad R, Chowdhury R. "Adaptive intrusion detection based on boosting and Naive Bayesian classifier", International Journal of Computer Application 2011;24(3):12–9.

[2] Yang L. "The research of Bayesian classifier algorithms in intrusion detection system", IEEE international conference on E-Business and E-Government, ICEE 2010. Guangzhou, China; May 2010. pp. 2174–8.

[3] Hari O, Aritra K. "A hybrid system for reducing the false alarm rate of anomaly intrusion detection system", IEEE international conference on recent advances in information technology, RAIT 2012. Dhanbad, India; March 2012. pp. 131–6.

[4] Alexander T, Aleksey P, So Grigory. "Efficient computer network anomaly detection by change point detection methods". IEEE J Sel Top Signal Process 2013,7(1):4–11.

[5] Sumaiya T, Aswani C. "An analysis of supervised tree based classifiers for intrusion detection system", IEEE proceedings of the international conference on pattern recognition, informatics and mobile engineering, PRIME 2013. Salem, India; February 2013. pp. 294–9.

[6] Chirag M, Dhiren P." Bayesian classifier and snort based network intrusion detection system in cloud computing", The third IEEE international conference on computing communication & networking technologies, ICCCNT 2012. Coimbatore, India; July 2012. pp. 1–7.

[7] S. Roschke, C. Feng and C. Meinel," An Extensible and Virtualization Compartible IDS Management Architecture," Fifth International Conference on information Assurance and Security, vol, 2,2009, pp. 130-134.

[8] Wafa .AL-Sarafat, and Reyadh Naoum" Development of Genetic –based Machine Learning for Network Intrusion Detection "World Academy of Science, Engineering and Technology 55,2009.

[9] D. Nurmi, R. Wolski, C.Grzegorczyk, G. Obertelli, S. Soman, L. Youseff and D. Zagordnov. (2008) "Eucalyptus: A Technical Report on an Elastric Utility Computing Architecture Linking Your Programs to Useful Systems", UCSB Computer Science Technical Report Number 2008-10.

[10] Nabil A, Soroush H, Ljiljana T. "Feature selection for classification of BGP anomalies using Bayesian models", Proceedings of the international conference on machine learning and cybernetics. ICMLC 2012. Xian, Shaanxi, China; July 2012. pp. 140–7.

[11] Peng H, Fulmi L, Ding C. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and minredundancy", IEEE Trans Pattern Anal Mach Intell 2005;27(8):1226–38.

[12] Chandrasekhar M, Raghuveer K. Intrusion detection technique by using k-means, fuzzy neural network and SVM classifiers. In: IEEE international conference on computer communication and informatics. Coimbatore, India; January 2013. pp. 1–7.

[13] Mohamed M. Abd-Eldayem, "A proposed HTTP service based IDS", Egyptian Informatics Journal (2014) 15, 13–24.

[14] The NSL-KDD data set http://nsl.cs.unb.ca/NSL-KDD/.

Author Profile

Sudhansu Ranjan Lenka,
Department of Computer Science & Engineering,
Asst. Professor ,Trident Academy of Technology,
Email: sudhansulenka2000@gmail.com

Dr. Bikram Keshari Rath,
Department of Computer Science & Application,
Professor,Utkal University, Bhubaneswar, India