Classification of Cloud Data using Bayesian Classification

Krunal Patel[1], Rohit Srivastava[2]

[1]ME (Computer Science & engineering) Student, Parul Institute of Technology, Gujarat Technological University, Vaghodia Road, Limda, Vadodara, Gujarat, India

[2]Assistant professor, Parul Institute of Engineering and Technology, Gujarat Technological University, Vaghodia Road, Limda, Vadodara, Gujarat, India

Abstract: ***One of the major security challenges in cloud computing is the detection and prevention of intrusions and attacks. In order to detect and prevent malicious activities at the network layer, we propose a security framework which integrates a network intrusion detection system (NIDS) in the Cloud infrastructure. We use snort and Bayesian classifier machine learning based techniques to implement this framework. To validate our approach, we evaluate the performance and detection efficiency of our NIDS by using KDD experimental intrusion datasets. The results show that the proposed model has a higher detection rate with low false positives at an affordable computational cost.***

Keywords: Cloud computing; Firewall; Intrusion detection system; Snort; Bayesian Classifier.

1. Introduction

Cloud computing nowadays is growing rapidly and is an innovative computing model that delivers convenient, on-demand, network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications etc) "as a service" on the Internet for satisfying computing demand of users. As shown in Fig. 1, it delivers services in various forms: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as Service (IaaS) [1].

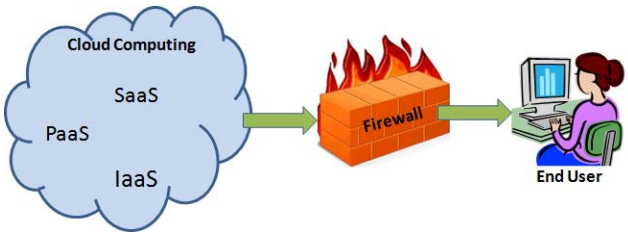

Figure 1: General architecture of Cloud computing.

As Cloud services are delivered through the Internet; security and privacy of Cloud resources and offered services are the biggest concerns. International Data Corporation (IDC) survey showed that security is the greatest challenge of Cloud. L. Martin Cyber Security division [2] shows that the major security concern after data security is intrusion detection and prevention in cloud.

At network layer, Cloud suffers from traditional attacks such as IP spoofing, Address Resolution Protocol (ARP) spoofing, Routing Information Protocol (RIP) attack, DNS poisoning, man-in-the-middle attack, port scanning, Insider attack, Denial of Service (DoS), Distributed Denial of Service (DDoS) etc. E.g., an internal DoS attack demonstrated against the Amazon Elastic Compute Cloud (EC2) [3]. DoS attack on the underlying Amazon Cloud caused BitBucket.org, a site hosted on Amazon Web Services (AWS) to remain unavailable for few hours [4]. These attacks affect the confidentiality, integrity and availability of Cloud resources and offered services. To address such issues, major Cloud providers (like Amazon, Window Azure, Rack Space, Eucalyptus, Open Nebula etc.) use the firewall, as shown in Fig 1. Firewall protects the front access points of system and is treated as the first line of defense. As firewall sniffs the network packets only at the boundary of a network, insider attacks cannot be detected. Few DoS or DDoS attacks are too complex to detect using traditional firewall. E.g., if there is an attack on port 80 (HTTP server) or port 25 (Mail server), firewall cannot differentiate normal traffic from attack traffic [5].

Another solution is to integrate network based intrusion detection system (NIDS) in Cloud computing. NIDS performs the role of alert system and adds the next preventive layer of security by detecting network attacks that penetrate our system. There are two types of techniques used in NIDS. One is signature based detection that can detect known attacks efficiently and another is anomaly detection that determines whether a given behavior is malicious or not. The efficiency of NIDS depends on parameters like used detection technique (signature based or anomaly based), it's positioning within network (front end or back end), its configuration (centralized or distributed).

In this paper, we design and integrate NIDS module in the Cloud (offering Infrastructure as a Service-IaaS) [5] to detect network attacks. In this module, we use snort and Bayesian classifier [6] techniques. Snort is used to detect known attacks, whereas Bayesian classifier predicts if the given event is malicious or not by observing previously stored network events. Our NIDS module ensures low false positives and high detection accuracy with affordable computational cost.

The rest of this paper is organized as follows: Section 2 presents existing NIDS approaches in Cloud followed by theoretical background to Bayesian Classifier and snort used

in our NIDS module. A detailed description of the proposed framework is given in section 4. Experimental results related to the performance and quality of proposed NIDS, are given in section 5. Section 6 concludes our work with references at the end.

2. Related Work

There have been several work to date to detect intrusions in the Cloud. A. Bakshi et al. [7] proposed an approach to detect a DDoS attack in a VM. An IDS is installed in the virtual switch to log incoming or outgoing traffic into the database. To detect known attacks, the logged packets are analyzed and compared with known signature in real time. An IDS determines the nature of attacks and notifies the virtual server. Then virtual server drops packets coming from the specified IP address. If attack type is DDoS, all the zombie machines are blocked. Then virtual server transfers targeted applications to other machines hosted by the separate data center and routing tables are immediately updated. Firewall placed on new server blocks all the packets coming from an identified IP address. This approach can block the DDoS attack in a virtual environment. However, it cannot detect all types of attacks as the tool used here (i.e. Snort) identifies only known attacks.

C. Mazzariello et al. [8] presented Snort based misuse detection in open source Eucalyptus Cloud. In this approach, Snort is deployed at the cloud controller as well as on the physical machines (hosting virtual machines) to detect intrusions coming from external networks. This approach solves the problem of deploying multiple instances of IDS as in. Although it is a fast and cost effective solution, it can only detect known attacks since only Snort [9] is involved.

Sandar et al. [10] introduced a new type of DDoS attack, called Economic Denial of Sustainability (EDoS) in Cloud services and proposed a solution framework for EDoS protection. EDoS attack can be called as HTTP and XML based DDoS attack. EDoS protection framework uses firewall and puzzle server to detect EDoS attack. A firewall is used to detect EDoS at the entry point of Cloud, whereas the puzzle server is used to authenticate the user. In this work, the authors demonstrated EDoS attack in the Amazon EC2 Cloud. However, it is not an efficient solution since it uses only traditional firewalls. Research is still needed to detect EDoS attacks in the Cloud.

A.V. Dastjerdi et al. [11] proposed scalable, flexible and cost effective method to detect intrusion for Cloud applications regardless of their locations using mobile agents. This method aims to protect VMs that are outside the organization. Mobile agent collects evidences of an attack from all the attacked VMs for further analysis and auditing. This approach is used to detect intrusion in VMs migrated outside the organization. However, it produces more network load, if the numbers of VMs attached to the mobile agent increases.

The proposed framework provides a novel solution that can overcome some of the limitations discussed above in the existing approaches. In the following section, we discuss a short theoretical background on Snort and Bayesian classifier used in our NIDS module.

3. Theoretical Background

Some of the Theoretical background is presented below.

3.1 Snort

For signature based detection, we use Snort [9], a well-known open source packet sniffer and NIDS. It is configurable, free, can run on multiple platforms (i.e. GNU/Linux, Window) and is constantly updated. It captures network data packet and checks their content with the predefined patterns for any correlation. The detection engine of Snort allows registering, alerting and responding to any known attack. Snort with firewall [12] can be used as an intrusion prevention system. In inline mode of Snort, the functionality of Snort is extended that provides active defense capability. However it cannot detect unknown attacks. Therefore, we use Bayesian classifier, a machine learning approach.

3.2 Bayesian Classifier [6]

We use naïve Bayes' classifier that is a statistical classifier to predict the probability of a given network event belong to a particular class (normal or intrusion). It has higher accuracy and speed than the other classifiers (e.g. ANN classifier) [6].

Let X is a given packet. H is hypothesis, such that X belongs to class C. We need to determine the probability P(H|X), that the hypothesis holds the packet X. P(H) is the initial probability of H. P(X) is the probability that a packet is observed. P(X|H), the probability of observing packet X, given that the hypothesis holds. Using Bayes' theorem, the probability P(H|X) of a hypothesis H, on given packet X can be derived by

$$P(H/X)=(P(X/H)P(H))/P(X)\text{……...(1).}$$

Bayesian classifier works as follows: let D is a training set of packets and their related class labels. Each packet is represented by a vector X = (x1, x2… xn). Suppose there are m different classes C1, C2… Cm. Bayesian classifier predicts X to class Ci, if the probability P (Ci|X) is the highest among all the P(Ck|X), for all the k classes. So, classification is to derive the maximal P(Ci|X). It can be derived from

$$P(Ci/X)=(P(X/Ci)P(Ci))/P(X)\text{…….(2).}$$

Since P(X) is constant for all classes, we need to maximize

$$P(H/X)=P(X/Ci)P(Ci)\text{……..(3).}$$

Thus, Bayesian classifier predicts the class label (normal or intrusion) by observing previously stored events.

4. Framework for NIDS in Cloud

4.1 Threat Model for Cloud

Cloud clients are facing two types of security threats viz; external and internal attacks as shown in fig. 2.

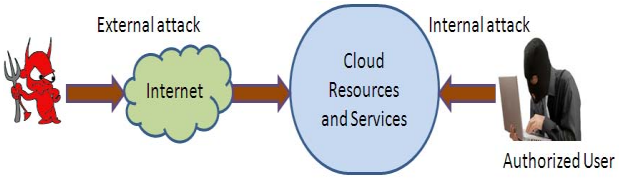

Figure 2: Threat model for Cloud computing.

External network attacks in the cloud are increasing at a notable rate. Malicious user outside the Cloud often performs DoS or DDoS attacks to affect the availability of Cloud services and resources. Port scanning, IP spoofing, DNS poisoning, phishing are also executed to gain access of Cloud resources.

Internal attacker (authorized user) can easily get access to other user's resources without being detected. An insider has higher privileges and knowledge (related to network, security mechanism and resources to attack) than the external attacker. Therefore, it is easy for an insider to penetrate an attack than external attackers.

4.2 Design Objectives

We expect the following goals from our proposed framework:

- Detection of network intrusions in Cloud
- Low false positives and false negatives
- High accuracy
- Low computational cost and faster detection rate
- Scalability
- Compatibility

4.3 Integration of NIDS in Cloud

Cloud computing can be viewed with two ends viz; front-end (user side) and back-end (processing server side). As shown in Fig. 3, front-end is connected to the external network and an internal network (virtual private network). Cloud users are able to communicate with Cloud via front end and request the instances of offered services through external network (Internet). Processing server consists of computer hardware and software that are designed for the delivery of services, including multi-core processors, cloud-specific operating systems and combined offerings.

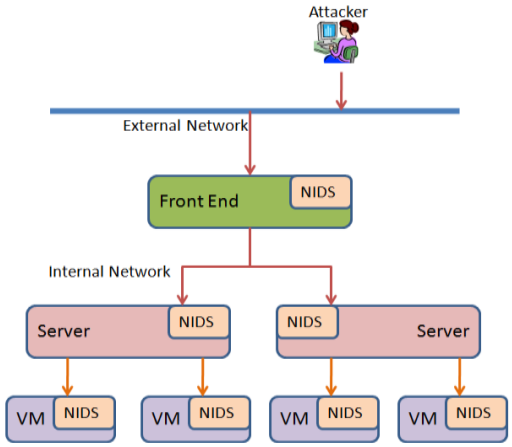

Figure 3: Integration of NIDS in Cloud.

Internal network (virtual network) is designed for VM instance interconnectivity. E.g., In Amazon cloud, each VM instance has two networks IPs named public IP and private IP [13]. VMs can communicate directly using private network. Network Address Translation (NAT) maps public IP of VM to the private IP of VM.

There may be different positioning of NIDS in Cloud, as (1) On front-end (2) On back-end (or processing server) and (3) On each virtual machine (VM). Pros and cons of each positioning are as follows:

Integrating NIDS module at front-end of Cloud helps to detect network intrusions at external network. However, it is not able to detect insider attacks.

Positioning NIDS module on back-end (processing server) helps to detect intrusions at internal network of Cloud. It will also detect the intrusions coming from external network. If integrated NIDS has faster detection capability, we recommend integration of NIDS behind the virtual termination point to monitor external and internal network events. This gives more opportunity to detect intrusions due to high traffic rate at processing server.

Integrating NIDS module on each VM helps the user to detect intrusions on his/her VM. Such configuration requires multiple instances of NIDS, which makes complex management of NIDS since VMs are dynamically migrated.

4.4 Proposed Architecture

As shown in Fig. 4, we propose a NIDS module consisting of four main components viz; packet preprocessing, intrusion detection (signature based detection and anomaly detection), storage (knowledge base, behavior base and central log) and alert system.

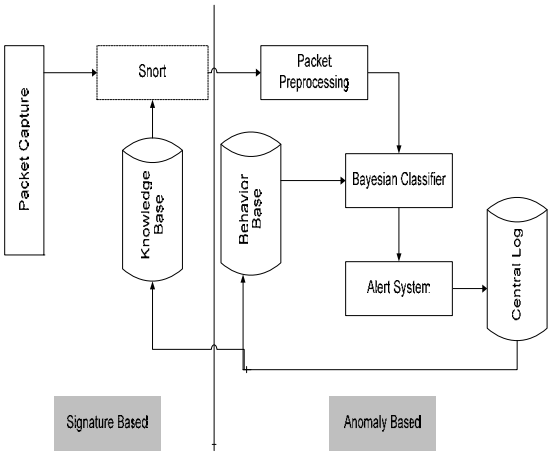

Figure 4: Proposed NIDS module.

The packet capture module is used to capture network packets, where it resides. Packet preprocessing module preprocesses the network packet by removing some redundant information that has very low correlation with detection.

Intrusion detection module consists of Snort and Bayesian classifier. Snort is used to detect known attacks by correlating captured packets with rules in the knowledge base. Bayesian classifier predicts class label of preprocessed packets.

Storage consists of three databases viz; knowledge base, behavior base and central log database. Knowledge base stores known attack signatures, where as behavior base stores network behavior having malicious packets and normal packets. Central log is used to log malicious event that is considered either by Snort or Bayesian classifier. NIDS modules deployed on other servers update their knowledge base and behavior base, if central log is updated.

Alert system is used to generate alert, if any anomaly is detected by Snort or Bayesian classifier.

As shown in Fig. 5, network packets are captured from external or internal networks. Snort matches the captured packets with the rules stored in the knowledge base. If any correlation is found, an alert is generated and stored in central log. Normal packets are preprocessed and applied to Bayesian classifier that predicts class label of such packets. Bayesian is trained using behavior base. If Bayesian classifier finds any intrusion, it will be alerted and stored in the central log. Otherwise, Bayesian considers those packets as normal events.

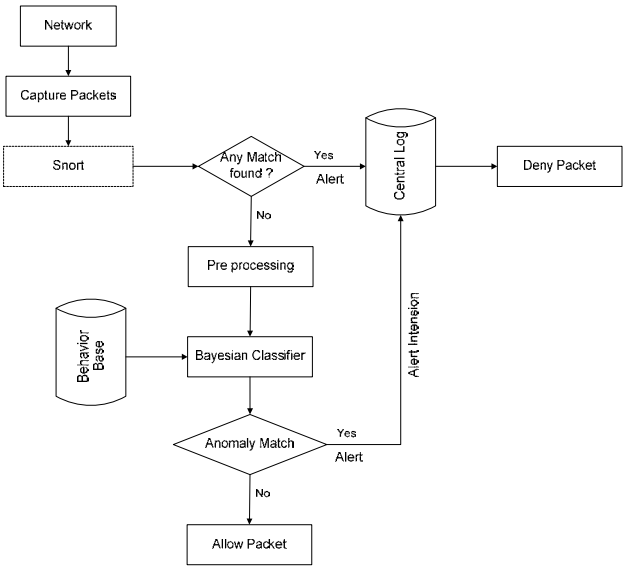

Figure 5: Workflow of NIDS module.

Combining signature based detection and anomaly detection in our NIDS module improves detection accuracy; since they are complimenting each other. Moreover, the signature based detection technique is applied prior to anomaly detection, which reduces the computational cost. Bayesian classifier has to detect only unknown attacks, because known attacks are already detected by Snort and denied. NIDS placed on all the servers update their knowledge base by getting alerts stored in central log. So, any unknown attack (that was previously detected at any server) can be easily detected by Snort at other servers. This also helps to reduce computational costs for detecting intrusions at other servers.

Following table 1 is showing the example training data packets that demonstrates Bayesian classifier in the proposed NIDS module.

Table 1: Sample Training Packets

ID	*Protocol Types*	*Service*	*Flag*	*Land*	*Class*
1	TCP	HTTP	S1	0	Normal
2	TCP	HTTP	S1	1	Normal
3	UDP	HTTP	S1	0	Intrusion
4	ICMP	SMTP	S1	0	Intrusion
5	ICMP	FTP	S0	0	Intrusion
6	ICMP	FTP	S0	1	Normal
7	UDP	FTP	S0	1	Intrusion
8	TCP	SMTP	S1	0	Normal
9	TCP	FTP	S0	0	Intrusion
10	ICMP	SMTP	S0	0	Intrusion
11	TCP	SMTP	S0	1	Intrusion
12	UDP	SMTP	S1	1	Intrusion
13	UDP	HTTP	S0	0	Intrusion
14	ICMP	SMTP	S1	1	Normal

As discussed earlier, Bayesian algorithm first finds the best feature (that has higher information gain) to Classify the data using equations (1), (2) and (3) as presented in section 3.2. The feature selection procedure is based on the Information gain of particular feature. The relative information gain is depicted in section 5.

5. Evaluation

5.1 Experimental Setup

We have used VMware [13] cloud installed on the Ubuntu operating system. As shown in Fig. 6, NIDS module that we have proposed can be installed on any one of the VM . we have to Enable the Promescous Mode of the VM in which the NIDS has been installed . So, the virtual pipe type phenomenon is created and the traffic in any other VM is also goes to the VM in which the NIDS is installed. Thus, evry network Traffic will be passing through this one VM. So, we can monitor evry network activity and capture the packets which go to any of the VM.

For testing purpose, we allow all types of traffic by opening all the ports in VMware. We have used Scapy [14] for sending custom packets on the network. Wireshark (WS) [15] installed on the front end and back end of Cloud to monitor traffic. MySql database (DB) is used to log intrusions.

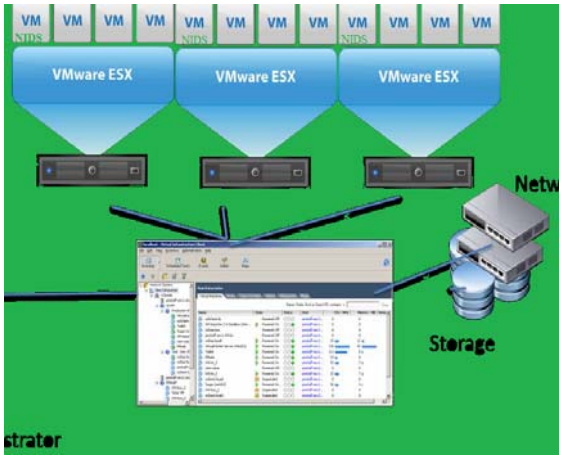

Figure 6: Proposed NIDS in VMware Cloud.

We have used 10% KDD'99 intrusion detection dataset [16] as training data for the Bayesian classifier. Details of dataset are given in Table 2.

Table 2: Details of KDD Dataset

Dataset	*No. of training data*	*No. of test data*	*No of features*	*No. of distinct classes*
KDD	4,94,021	3,11,029	41	25 (24 attack types + 1 Normal)

We have preprocessed class labels of training data and converted into two distinct classes viz; Normal and Intrusion. We have used 17 features out of 41 features since classification on 41 features of KDD'99 dataset decreases detection accuracy [17]. As shown in Table 3, selected features have better information gain to improve detection accuracy. We preprocessed continuous valued features (src_bytes, dst_bytes, count and srv_count) and normalized into 100 blocks of size 500.

We have calculated the true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), false negative rate (FNR), fitness score, accuracy and computational cost [18], [19] to evaluate the feasibility of the proposed NIDS module in the Cloud.

Table 3: Information Gain of Selected Features in KDD Dataset

Feature No.	*Feature Name*	*Information Gain in KDD dataset*
2	protocol type	0.3024
3	Service	0.5709
4	Flag	0.0630
5	src_bytes	0.6460
6	dst_bytes	0.5383
7	Land	0.0005
8	wrong_fragment	0.0008
10	Hot	0.0029
11	num_failed_logins	0.0008
14	root_shell	0.0004
22	is_guest_login	0.0007
23	Count	0.6193
24	srv_count	0.3457
28	srv_rerror_rate	0.0023
30	diff_srv_rate	0.0831
36	dst_host_same_src_port_rate	0.3847
39	dst_host_srv_serror_rate	0.0801

6. Results & Discussions

Performance results are shown in Table 4. It shows that proposed NIDS has better precision value (> 95%) on KDD dataset, which indicates that more than 95% of intrusions are detected correctly. This shows that the 96.25% of legitimate intrusions are alerted as Intrusions.

Table 4: Results of Proposed NIDS Module

Dataset	*Precision*	*Recall/TPR*	*TNR*	*FPR*	*FNR*
KDD	99.64	96.09	98.96	1.04	3.91

Accuracy	*F_Score*
96.82	0.97

In general, our NIDS module is capable of detecting higher number of intrusions with low false positives and false negatives for the cloud environment. Moreover, it is scalable since new rules can be easily added into the Snort configuration file without modifying existing rules. Also, NIDS modules can be added or removed at VM without modifying the existing ones if separate NIDS module has been used. The proposed NIDS module is compatible with any communication protocol and platforms like Windows and Linux.

As discussed earlier, existing approaches [7], [8] are using only signature based technique. Hence, they can only detect known attacks, whereas our NIDS module detects known and unknown attacks, while satisfying the network security needs.

7. Conclusion

Security in Cloud computing is a major concern which is slowing the intake of cloud by corporate. Traditional firewall

fails to offer network security needs in the Cloud. In this paper, we have designed and incorporated a network based intrusion detection system (NIDS) that combines Snort and Bayesian classifier methods to detect known as well as unknown attacks in the Cloud environment (i.e. IaaS). We have used the signature based and anomaly based techniques to improve detection accuracy. In the proposed framework, Snort is applied prior to the Bayesian classifier to reduce the detection time. Experimental results show that proposed NIDS has a high detection rate with low false alerts and affordable computational cost. Moreover, it has high detection accuracy and F_score.

References

[1] Biggs, S., Vidalis, S.: Cloud computing: The impact on digital forensic investigations. International Conference for Internet Technology and Secured Transactions, ICITST 2009, pp. 1–6 (2009)

[2] Martin, L.: Awareness, Trust and Security to Shape Government Cloud Adoption, White Paper, (2010), http://www.lockheedmartin.com/content/dam/lockheed/data/corporate/ documents/Cloud-Computing-White-Paper.pdf

[3] Cloud Computing Comparison Guide, Web hosting unleashed http://www.webhostingunleashed.com/whitepaper/cloud-computing-comparison/

[4] Cybercrime Battle Basics: Online Account, Transaction and Device Protection, White Paper, (2012), www.threatmetrix.com/docs/Whitepaper-Cybercrime-Defender.pdf

[5] Mell, P., Grance, T: The nist definition of cloud computing(draft)-(2011), http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf

[6] Han, J., Kamber, M.: Data Mining Concepts and Techniques, 2nd edition, Morgan Kaufmann Publishers (2006)

[7] Bakshi, A., Yogesh, B.: Securing Cloud from DDOS Attacks Using Intrusion Detection System in Virtual Machine, Second International Conference on Communication Software and Networks, pp. 260-264 (2010)

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

[9] Snort-Home page, https://www.snort.org/

[10] S. V. Sandar, S. Shenai, Economic Denial of Sustainability (EDoS) in Cloud Services using HTTP and XML based DDoS Attacks, International Journal of Computer Applications, 41 (20), pp. 11-16 (2012).

[11] Dastjerdi, A. V., Bakar, K. A., Tabatabaei, S. G. H.: Distributed intrusion detection in clouds using mobile agents, in Third International Conference on Advanced Engineering Computing and Applications in Sciences, ADVCOMP ’09, pp. 175 – 180 (2009)

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

[13] S. Ram, “Secure cloud computing based on mutual intrusion detection system,” International journal of computer application, vol. 2, no. 1, 2012, pp. 57-67.

[14] Scapy, http://www.secdev.org/projects/scapy/

[15] Wireshark, http://www.wireshark.org/

[16] KDDcup,1999,http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[17] Sathya, S. S., Ramani, R. G., Sivaselvi, K.: Discriminant Analysis based Feature Selection in KDD Intrusion Dataset, International Journal of Computer Applications (IJCA), 31 (11), pp. 1-7, (2011)

[18] Sen, S., Clark, J. A., Tapiador, J. E.: Power-Aware Intrusion Detection on Mobile Ad Hoc Neworks, In Proc. of the first International Conference on Ad hoc Networks (2009)

[19] Thomas, C., Balakrishnan, and N. Performance enhancement of Intrusion Detection System using advances in sensor fusión, 11th International Conference on Information Fusion.pp.1-7 (2008)

Author Profile

Krunal V. Patelreceived the B.E degree in Computer Engineering from the Sardar Patel University, V.V.Nagar, Aanand in 2007 and M.E. degree in Computer Science & Engineering (currently pursuing) from Gujarat Technological University in 2013.The Area of interest is artificial intelligence, mobile computing, security, cloud computing.