Layered architecture for DoS attack detection system by combine approach of Naive bayes and Improved K-means Clustering Algorithm

Mangesh Salunke[1], Ruhi Kabra[2], Ashish Kumar[3]

[1] PG Student, Computer Eng, GHRCEM,SPPU, Maharashtra, INDIA
[2] Asst Professor, Computer Eng, GHRCEM,SPPU, Maharashtra, INDIA
[3] Asst Professor, Computer Eng, GHRCEM,SPPU, Maharashtra, INDIA

---***---

Abstract - *The aim of a DoS attack is to consume the resources of a victim or the resources on the way to* ***communicate with a victim. By wasting the victim's*** *resources, the attacker disallows it from serving legitimate customers. A victim can be a host, server, router, or any computing entity connected to the network. DoS attack can cause harm to these computer and network services. Therefore, effective detection of DoS attacks is essential to the protection of network and resources. Detection System is built by using layered frame work approach for an effective attack detection system. The proposed system will create own data set by analyzing the incoming packets in real time system, by comparing with previous existing system that uses Knowledge Discovery & Data Mining(KDD) 1999 dataset, and classify the DoS Attack such as SYN Flood, Ping Flood, UDP Flood.*

Key Words: Network Security; DoS Attack; DoS detection system; Naïve Bayes; K-means clustering; Real time IDS

1. INTRODUCTION

Attack is nothing but a Violation of security policy of system. There can be possibly two types of attack, Active attack, in which contains of original message are modified by attacker. Passive attack, in which attacker only aims to gain the transit information.

Attacks can be classified as below:

Network based attack: These types of attacks are launched from a device other than those under attack. In this attacker uses one or more devices to overload the server with so much traffic so that server cannot respond to authorize user's request.

Host based attack:Attackers exploit vulnerabilities of system and application to launch the DoS attack. These types of attacks are application specific, i.e., exploiting algorithms, memory structure, authentication protocols etc., which makes it different from network based attack.

The traffic of host based attacks may not be as high as network based attacks, because application flaws and deficiencies can easily crash applications or consume a tremendous amount of computer resources

1.1 Security goals

-Confidentiality: Hiding transmitted data from unauthorized users.
-Integrity: Preventing transmitted data from unauthorized modification.
-Availability: Ensures for authorize user the data or system is always available. [1]

.

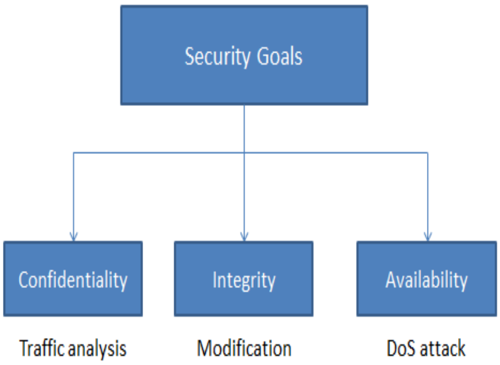


Fig1: Goals of security and treats

1.2 DoS Attack

It's nothing but Denial of service attack. As name suggest the attacker prevents or deny the service of the authorize user. Attacker prevents the access of system or resources to be used to its authorize user. The main goal or aim of DoS attack is to disturb the activity of authorize user that may be accessing server, some resources, browsing web pages, accessing social networking sites etc. DoS attack can be perform in two ways such as one way is attacker crash the services and in another way attacker sends vast amount of traffic to consume the resources, in both ways all of the targets critical resources are busy to handling the attack traffic therefore they are unavailable to authorize user [2].

Aims of DoS attack are:
-Consuming the bandwidth by sending large volume traffic.
-Consume limited available resources by sending specific type of packets.
-Flooding packets to crash or overload the network

1.3 Types of DoS Attack

1: TCP SYN Flood: In this type of attack, attacker sends vast amount of SYN packet request by using spoofed sender IP address to establish a TCP connection with the server. Upon receiving the SYN request the server sends the TCP-SYN ACK to that spoofed IP address and goes on wait state for last ACK which never comes back because the IP address is spoofed and before this waiting time expires the attacker sends another SYN request to the server and this process is continue. Because of this the authorize user cannot access the server [3]

2: PING Flood: In Ping flood attack attacker uses Ping command to perform the DoS attack which can be known as ICMP Ping attack or simply Ping attack. This is very simple type of DoS attack in which attacker sends vast amount of ping packets that is ICMP ECHO request packet to the target system, the target system responded with ICMP ECHO reply to each request. Because of this continuous request-response the target system may get slow or some time may get crash.[3]

3: UDP Flood: In this type of attack, attacker simply sends or floods the UDP packets containing IP packets on random ports of the target system. The target system opens the packet and found nothing in it and send back destination unreachable packet. Because of this vast amount of UDP packet traffic the target system or the resources at the target system can be busy for serving these request and resulting in unavailable to the authorize users [3]

1.4 Why should we care?

CSI/FBI report on Computer Crime and Security in 2012, the DoS attack faced by 32% of respondents. DoS attacks are the 2nd most costly form of attacks. DoS attacks carry six figure price tag for businesses, costs large businesses an average of $444,000 in lost revenue. Overall, nearly 1 in 5 businesses experienced a DoS attack during the year-long study period. DoS attack becomes one of the most threatening attack to security of computer networks. Also the use of Internet today is increasing, the study shows that DoS attack has highest ranking among other attacks, so there is need to counter such type of attack.[4]

1.5 DoS Countermeasures techniques

1: Attack prevention system: DoS attack prevention system is used before the attack happen and to reduce attack attempts without preventing the user to use services by providing backup services available. This technique can be preferred approach to DoS attack but may be impractical with all types of DoS flooding attacks.
2: Detection system: DoS attack detection system is used during the attack to detect the attack to minimizing the impact of attack. It detects suspicious pattern or behavior of that packet. [2]

The rest of the paper is organized as follows. Section 2 describes the Literature survey. Section 3 describes the proposed system architecture and basic terminology used for proposed system. Result shown in the section 4

2. RELATED WORK

Aikaterini Mitrokotsa et al. proposed an approach in [4] detects Denial of Service attacks using Emergent SOM that is based on classification of traffic from normal to abnormal traffic. There experiment shows result for detection rate for DoS attacks that ranges between 98,3% and 99,81% but they still have the false alarm rate that ranges between 2.9% to 0.1%. Kemal Bicakci et al. present in [5] systematic survey of DoS attacks, which exploits MAC and physical layer vulnerabilities of 802.11 networks. Peng Ning et al. discuss in [6] specific type of DoS attacks, and classify the different attacking patterns, also presented a dynamic window scheme that can effectively contain the damage of DoS attacks to a small portion of the nodes. The results of the experiments shows that updating window size based on the validity history of the incoming messages, is the best in terms of containing DoS attacks. But it requires higher cost because sensor nodes need to remember more information. Dapeng Wu et al. propose in [7] a novel framework that not only efficiently detect DoS attacks but also identify packets that are used for Attack. In this framework the traffic is analyzed only at the edge routers of an ISP network. There experimental results show that the proposed framework can detect DoS attacks for large traffic and has a detection probability of 0.97. Tao Peng et al. analyze in [8] the design decisions in the Internet that have created the potential for denial of service attacks and also the state-of-art mechanisms for defending against denial of service attacks. Mark Handley el al. present in [9] they most focus on flooding DoS attacks prevention, and defend the remaining attacks. They have outlined a set of changes to the Internet architecture including the explicit separation of the IP address space into client and server addresses, along with associated rules restricting how those addresses can be used. Xiangjian He et al. proposed in [10] DoS attack detection system, that is Earth Mover's Distance (EMD) developed by useing dissimilarity measure. It accepts cross bin matching as input and provides accurate evaluation, for that tenfold cross validations are conducted using KDD Cup 99 data set and ISCX 2012 IDS Evaluation data set The result of EMD DoS attack detection system 99.95% detection accuracy on KDD Cup 99 data set and 90.12% detection accuracy on ISCX 2012 IDS. Wanlei Zhou et al. propose in [11] generalized entropy metric the information distance metric for detection of low rate DoS attack. classify statistic IP packets and compute the probability distributions of the source IP addresses in attack and attack-free scenarios, For experiment, they use the MIT Lincoln Laboratory Scenario as the normal network traffic, and use the Low-rate DoS attack scenario from CAIDA. They outperform the traditional Shannon entropy and Kullback–Leibler distance approaches Mieso K. Denko et al. proposed in [12], a reputation-based incentive mechanism for detecting DoS attacks, that classified into trade-based and trust-based mechanisms. In the trust-based models, trust is created and the service provider is stimulated by these trust values. Each scheme can be deployed in different application scenarios. The trade-based models are not applicable in cooperative networks where no financial incentives are needed to run the network. There mechanism involves cluster formation, reputation database construction and maintenance, and information exchange. The result show that The overhead ranges between 14% and 25% and detection rate of selfish nodes increases from 80% to 99% with cluster-level reputation information and from 76% to 97% with neighbor level reputation information. Phyu Thi Htun et al explore in [13] the classification methods for Denial-of-Service (DoS) attacks detection using with Random Forests (RDF) and k-Nearest Neighbor by using KDD 99 data set. There system reduces the training time and also increases the accuracy of the system's classification. The experimental results shows that detection rate at 99.97%

3. PROPOSED SYSTEM

In today's world the use of Internet has increased so drastically. Also the DoS attacks are becoming real threat to internet, so to countermeasure them the first step is to detect such type of attacks

3.1Mathematical model For proposed system

1: NP-hard analysis:
In Naïve Bayes classification algorithm, the problem of belief propagation, that is, the calculation of probabilities at the hypothesis nodes when evidence is entered at information nodes – is, in general, NP-hard. Note also that the NP-hard calculations need to be done only once, given that the information and hypothesis nodes do not change. Our proposed solution takes advantage of this fact as the sets of information and hypothesis nodes remain static. This allows our system to analyze a stream of system calls in real-time without incurring noticeable computational or memory overhead. The number of samples to be processed is very high. Algorithms have to be very conscious of scaling issues. Like many interesting problems, clustering in general is NP-hard, and practical and successful data mining algorithms usually scale linear or log-linear.

2: Set theory:
Let, S = {I,O,U}
Where,
S is System.
I is set of Input.
Such that, I= {P, Fin}
P is set of Packets.
Such that, P = {P1, P2, P3, P4,..., Pn}
Fin is set of features.
Such that, Fin={length ,SRC_IP, DEST_IP, SRC_MAC,DEST_MAC,HOP_LIMIT, RST_FLAG,...}
O is set of output.
Such that, O= {Normal, Attack}
Such that, Attack={SYNFLOOD, UDPFLOOD, PINGFLOOD}

3: Functionalities: Improved K-means:
Input: D: The set of n tuples with attributes Al, A2, . . ., Am where m = no. of attributes. All attributes are numeric
Output: Suitable number of clusters with n tuples distributed properly
Compute sum of the attribute values of each tuple (to find the points in the data set which are farthest apart)
Take tuples with minimum and maximum values of the sum as initial centroids
Create initial partitions (clusters) using Euclidean distance between every tuple and the initial centroids

$$d(x, y) = \sum_{i=0}^{n} \sqrt{(xi - yi)^2}$$

Find distance of every tuple from the centroid in both the initial partitions. Take d=minimum of all distances.
Compute new means (centroids) for the partitions created in step 3.
Find the outliers depending on the following objective function:
If Distance of the tuple from the cluster mean < d then not an Outlier.
Compute new centroids of the clusters.
Calculate Euclidean distance of every outlier from the new cluster centroids and find the outliers not satisfying the objective function in step 5.
Let B= {Yl,Y2,. ....Yp) be the set of obtained outliers.
Repeat until (B== Φ)
Create a new cluster for the set B, by taking mean value of its members as centroid.
Find the outliers of this cluster
If no. of outliers = p then
Create a new cluster with one of the outliers as its member
Naive Bayesian Classifier:
Input: n No. of clusters
Output: classification of packet either Ok or Attack
By Bayesian theorem,
P (Ci| X)= P (X | Ci) P(Ci)/P(x)
As P(X) is constant for each classes, only P (X|Ci) P (Ci) need be maximized.
Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, X=(x1, x2, ... , xn), depicting n measurements made on the tuple from n attributes, respectively, A1, A2,.., An.
2) Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the naïve Bayesian classifier predicts that tuple x belongs to the class Ci if and only if
$P (Ci|X)>P (Cj|X)$ for $1\le j\le m, j \neq i$

We have to calculate the probability of each attribute with respect to selected features.
Calculate the probability of each attribute
i.e. P(x|ci)P(ci) for each i=1,2,3…
In order to predict the class label of X, P(X|Ci)P(Ci) is evaluated for each class Ci. The classifier predicts that the class label of tuple X is the class Ci if and only if
P(X|Ci)P(Ci)>P(X|Cj)P(Cj) for $1 \le j \le m, j \ne i$
In other words, the predicted class label is the class Ci for which P(X|Ci)P(Ci) is the maximum.

3.2 Proposed System architecture

Proposed system is implemented using divide and conquers methodology. That is architecture of proposed system is divided into two parts, Training set generation and Real time layered IDS. Again functionality of each module is divided into different modules of two basic modules of proposed architecture, such as packet sniffer, packet analyzer, feature extraction, feature selection and so on.
Two Basic Modules:
1: Training set generation:
In existing system [16], it uses KDD 99 dataset, but in the proposed system detection of attack is done by using real time data. For that the incoming packets towards system goes through each level of training set generation module. And finally stored as Ok or Attack in the database.

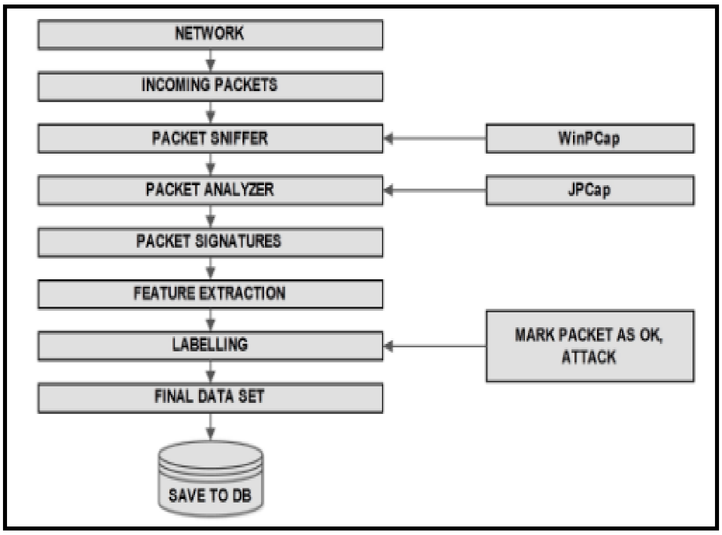
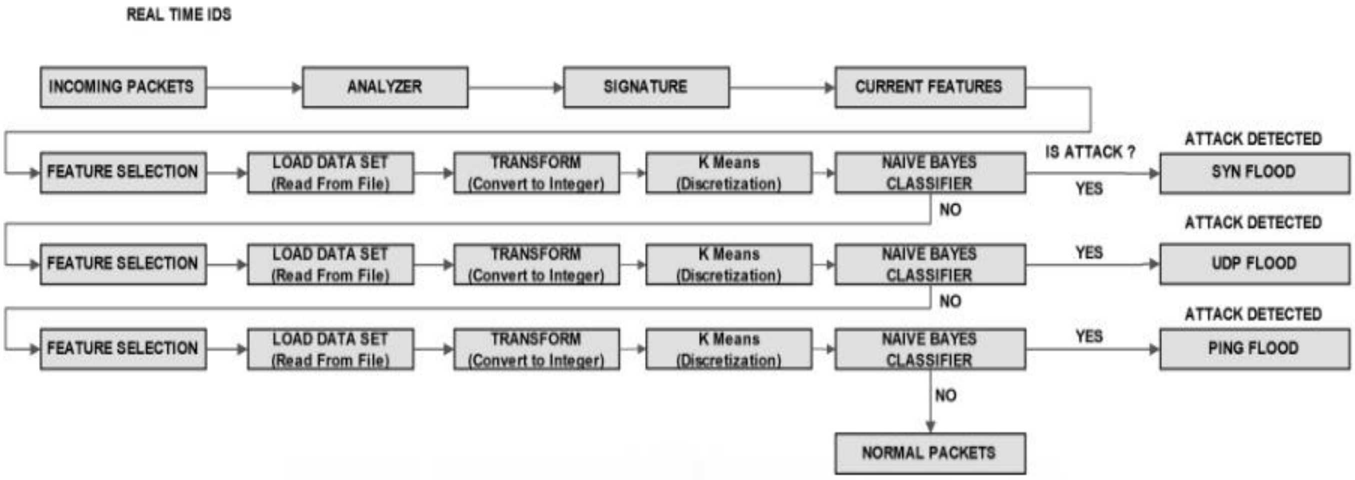


Fig -2: Training set generation



2: Real time layered IDS:
In real time layered IDS module, actual detection of attack is done. For classification of packet between attack or normal packet proposed system uses this layered module. In this module packet goes series of levels. First the signature of packet is capture by using signature module and according to that the features are extracted from incoming packets. Next according to selected features the data is loaded from Dataset and by using improved k-means algorithm and Naïve Bayes classification algorithm the classification is done

4. DATASET

The dataset is built by using Training set generation module. In that all the incoming packets are goes through series of steps and finally labeled as “Normal” or “Attack”, and stored into database for future use.

5. BASIC TERMINOLOGY USED

1: Feature selection:Feature selection is one of the most important preprocessing steps in data mining. Selection of useful and information bearing input features is vital for successful detection of DoS.

2: Improved k-Mean Clustering : It is modified version of K-means clustering algorithm. The improved k-means algorithm does not require number of clusters (K) as input. In this algorithm initially two clusters are created by choosing two initial Centroids which are farthest apart in the data set, so that in the initial step itself we can create two clusters with the data members, which are the most dissimilar ones.

Input: D: The set of n tuples with attributes Al,A2, . . . , Am where m = no. of attributes

Output: Suitable number of clusters [14]

3: *Naive Bayes:*

A naive Bayes classifier is a simple classifier in which a probability of given data set is found onto the given query. Naive Bayes is used most of the time for machine-learning and data mining methods. It uses when input is too high for classification of data. [15]

6. RESULT

6.1 Result Evaluation:

The performance of proposed system can be measured by using classical evaluation metric Precision and Recall. These metrics are traditionally defined for a binary classification task with positive and negative classes. Precision is the proportion of positive predictions that are correct, and recall is the proportion of positive samples that is the fraction of relevant instances that are retrieved

Table -1: Result Evaluation

Actual	Predicted	
	Normal	Attack
Normal	TN	FP
Attack	FN	TP

Where,

True Positive (TP): The number of the malicious packets correctly classified as malicious.

False Positive (FP): The number of normal traffic falsely classified as malicious.

False Negative (FN): It occurs when the malicious traffic is classified as normal traffic.

True Negative (TN): The number of benign packets correctly classified as benign.

The rate of accuracy, detection and false alarm is calculated by using,

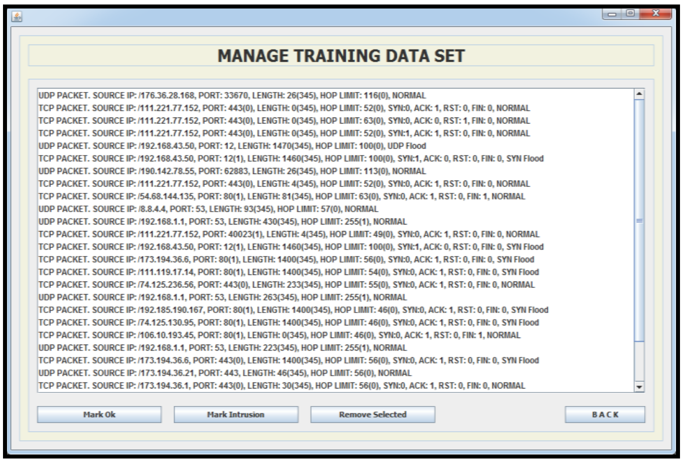
Accuracy = (TP+TN) / (TP+TN+FP+FN)*100

Detection rate =TP / (TP+FP)*100

False rate = FP / (FP+TN)*100

6.2. Clustering result:

Clustering that is improved k-means clustring algorithm is applied on Dataset (shown in fig. 4) and clusters are formed by using clustering module.Table II shows the result of clustering.The significance of this clustering result is very important from the point of view of classification of packet.As we got number of clusters as ouput of this module,we have to check next incoming packet and choose one of the cluster for that packet and accordingly we can select the features from that packet and classification can be done on that incoming packet

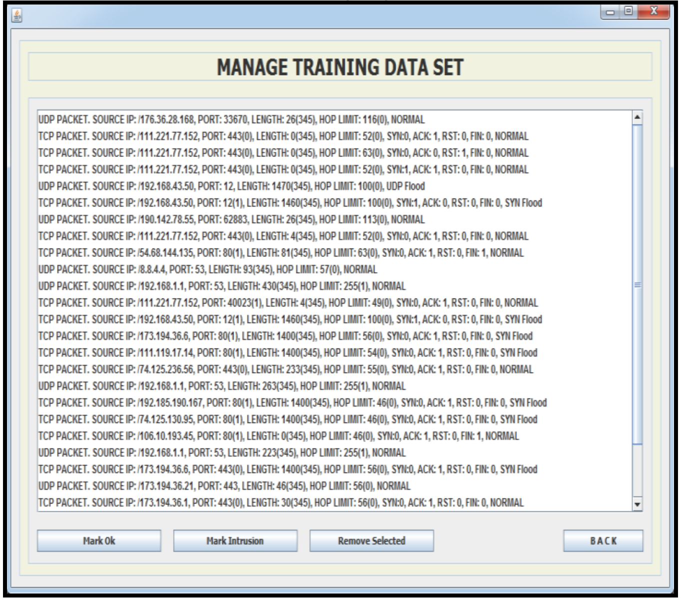


```
UDP PACKET. SOURCE IP: /176.36.28.168, PORT: 33670, LENGTH: 26(345), HOP LIMIT: 116(0), NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 443(0), LENGTH: 0(345), HOP LIMIT: 52(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 443(0), LENGTH: 0(345), HOP LIMIT: 63(0), SYN:0, ACK: 0, RST: 1, FIN: 0, NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 443(0), LENGTH: 0(345), HOP LIMIT: 52(0), SYN:1, ACK: 1, RST: 0, FIN: 0, NORMAL
UDP PACKET. SOURCE IP: /192.168.43.50, PORT: 12, LENGTH: 1470(345), HOP LIMIT: 100(0), UDP Flood
TCP PACKET. SOURCE IP: /192.168.43.50, PORT: 12(1), LENGTH: 1460(345), HOP LIMIT: 100(0), SYN:1, ACK: 0, RST: 0, FIN: 0, SYN Flood
UDP PACKET. SOURCE IP: /190.142.78.55, PORT: 62883, LENGTH: 26(345), HOP LIMIT: 113(0), NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 443(0), LENGTH: 4(345), HOP LIMIT: 52(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
TCP PACKET. SOURCE IP: /54.68.144.135, PORT: 80(1), LENGTH: 81(345), HOP LIMIT: 63(0), SYN:0, ACK: 1, RST: 0, FIN: 1, NORMAL
UDP PACKET. SOURCE IP: /8.8.4.4, PORT: 53, LENGTH: 93(345), HOP LIMIT: 57(0), NORMAL
UDP PACKET. SOURCE IP: /192.168.1.1, PORT: 53, LENGTH: 430(345), HOP LIMIT: 255(1), NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 40023(1), LENGTH: 4(345), HOP LIMIT: 49(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
TCP PACKET. SOURCE IP: /192.168.43.50, PORT: 12(1), LENGTH: 1460(345), HOP LIMIT: 100(0), SYN:1, ACK: 0, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /173.194.36.6, PORT: 80(1), LENGTH: 1400(345), HOP LIMIT: 56(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /111.119.17.14, PORT: 80(1), LENGTH: 1400(345), HOP LIMIT: 54(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /74.125.236.56, PORT: 443(0), LENGTH: 233(345), HOP LIMIT: 55(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
UDP PACKET. SOURCE IP: /192.168.1.1, PORT: 53, LENGTH: 263(345), HOP LIMIT: 255(1), NORMAL
TCP PACKET. SOURCE IP: /192.185.190.167, PORT: 80(1), LENGTH: 1400(345), HOP LIMIT: 46(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /74.125.130.95, PORT: 80(1), LENGTH: 1400(345), HOP LIMIT: 46(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /106.10.193.45, PORT: 80(1), LENGTH: 0(345), HOP LIMIT: 46(0), SYN:0, ACK: 1, RST: 0, FIN: 1, NORMAL
UDP PACKET. SOURCE IP: /192.168.1.1, PORT: 53, LENGTH: 223(345), HOP LIMIT: 255(1), NORMAL
TCP PACKET. SOURCE IP: /173.194.36.6, PORT: 443(0), LENGTH: 1400(345), HOP LIMIT: 56(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
UDP PACKET. SOURCE IP: /173.194.36.21, PORT: 443, LENGTH: 46(345), HOP LIMIT: 56(0), NORMAL
TCP PACKET. SOURCE IP: /173.194.36.1, PORT: 443(0), LENGTH: 30(345), HOP LIMIT: 56(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
```

Fig -4: Dataset

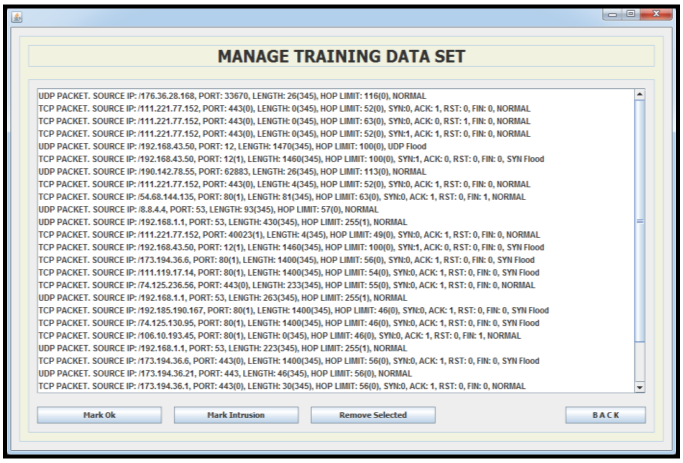
7. SCREENSHOTS

Some of screenshots from the project are shown below



```
UDP PACKET. SOURCE IP: /176.36.28.168, PORT: 33670, LENGTH: 26(345), HOP LIMIT: 116(0), NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 443(0), LENGTH: 0(345), HOP LIMIT: 52(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 443(0), LENGTH: 0(345), HOP LIMIT: 63(0), SYN:0, ACK: 0, RST: 1, FIN: 0, NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 443(0), LENGTH: 0(345), HOP LIMIT: 52(0), SYN:1, ACK: 1, RST: 0, FIN: 0, NORMAL
UDP PACKET. SOURCE IP: /192.168.43.50, PORT: 12, LENGTH: 1470(345), HOP LIMIT: 100(0), UDP Flood
TCP PACKET. SOURCE IP: /192.168.43.50, PORT: 12(1), LENGTH: 1460(345), HOP LIMIT: 100(0), SYN:1, ACK: 0, RST: 0, FIN: 0, SYN Flood
UDP PACKET. SOURCE IP: /190.142.78.55, PORT: 62883, LENGTH: 26(345), HOP LIMIT: 113(0), NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 443(0), LENGTH: 4(345), HOP LIMIT: 52(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
TCP PACKET. SOURCE IP: /54.68.144.135, PORT: 80(1), LENGTH: 81(345), HOP LIMIT: 63(0), SYN:0, ACK: 1, RST: 0, FIN: 1, NORMAL
UDP PACKET. SOURCE IP: /8.8.4.4, PORT: 53, LENGTH: 93(345), HOP LIMIT: 57(0), NORMAL
UDP PACKET. SOURCE IP: /192.168.1.1, PORT: 53, LENGTH: 430(345), HOP LIMIT: 255(1), NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 40023(1), LENGTH: 4(345), HOP LIMIT: 49(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
TCP PACKET. SOURCE IP: /192.168.43.50, PORT: 12(1), LENGTH: 1460(345), HOP LIMIT: 100(0), SYN:1, ACK: 0, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /173.194.36.6, PORT: 80(1), LENGTH: 1400(345), HOP LIMIT: 56(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /111.119.17.14, PORT: 80(1), LENGTH: 1400(345), HOP LIMIT: 54(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /74.125.236.56, PORT: 443(0), LENGTH: 233(345), HOP LIMIT: 55(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
UDP PACKET. SOURCE IP: /192.168.1.1, PORT: 53, LENGTH: 263(345), HOP LIMIT: 255(1), NORMAL
TCP PACKET. SOURCE IP: /192.185.190.167, PORT: 80(1), LENGTH: 1400(345), HOP LIMIT: 46(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /74.125.130.95, PORT: 80(1), LENGTH: 1400(345), HOP LIMIT: 46(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /106.10.193.45, PORT: 80(1), LENGTH: 0(345), HOP LIMIT: 46(0), SYN:0, ACK: 1, RST: 0, FIN: 1, NORMAL
UDP PACKET. SOURCE IP: /192.168.1.1, PORT: 53, LENGTH: 223(345), HOP LIMIT: 255(1), NORMAL
TCP PACKET. SOURCE IP: /173.194.36.6, PORT: 443(0), LENGTH: 1400(345), HOP LIMIT: 56(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
UDP PACKET. SOURCE IP: /173.194.36.21, PORT: 443, LENGTH: 46(345), HOP LIMIT: 56(0), NORMAL
TCP PACKET. SOURCE IP: /173.194.36.1, PORT: 443(0), LENGTH: 30(345), HOP LIMIT: 56(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
```

Fig -5: Capture Packets



MANAGE TRAINING DATA SET

```
UDP PACKET. SOURCE IP: /176.36.28.168, PORT: 33670, LENGTH: 26(345), HOP LIMIT: 116(0), NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 443(0), LENGTH: 0(345), HOP LIMIT: 52(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 443(0), LENGTH: 0(345), HOP LIMIT: 63(0), SYN:0, ACK: 0, RST: 1, FIN: 0, NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 443(0), LENGTH: 0(345), HOP LIMIT: 52(0), SYN:1, ACK: 1, RST: 0, FIN: 0, NORMAL
UDP PACKET. SOURCE IP: /192.168.43.50, PORT: 12, LENGTH: 1470(345), HOP LIMIT: 100(0), UDP Flood
TCP PACKET. SOURCE IP: /192.168.43.50, PORT: 12(1), LENGTH: 1460(345), HOP LIMIT: 100(0), SYN:1, ACK: 0, RST: 0, FIN: 0, SYN Flood
UDP PACKET. SOURCE IP: /190.142.78.55, PORT: 62883, LENGTH: 26(345), HOP LIMIT: 113(0), NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 443(0), LENGTH: 4(345), HOP LIMIT: 52(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
TCP PACKET. SOURCE IP: /54.68.144.135, PORT: 80(1), LENGTH: 81(345), HOP LIMIT: 63(0), SYN:0, ACK: 1, RST: 0, FIN: 1, NORMAL
UDP PACKET. SOURCE IP: /8.8.4.4, PORT: 53, LENGTH: 93(345), HOP LIMIT: 57(0), NORMAL
UDP PACKET. SOURCE IP: /192.168.1.1, PORT: 53, LENGTH: 430(345), HOP LIMIT: 255(1), NORMAL
TCP PACKET. SOURCE IP: /111.221.77.152, PORT: 40023(1), LENGTH: 4(345), HOP LIMIT: 49(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
TCP PACKET. SOURCE IP: /192.168.43.50, PORT: 12(1), LENGTH: 1460(345), HOP LIMIT: 100(0), SYN:1, ACK: 0, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /173.194.36.6, PORT: 80(1), LENGTH: 1400(345), HOP LIMIT: 56(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /111.119.17.14, PORT: 80(1), LENGTH: 1400(345), HOP LIMIT: 54(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /74.125.236.56, PORT: 443(0), LENGTH: 233(345), HOP LIMIT: 55(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
UDP PACKET. SOURCE IP: /192.168.1.1, PORT: 53, LENGTH: 263(345), HOP LIMIT: 255(1), NORMAL
TCP PACKET. SOURCE IP: /192.185.190.167, PORT: 80(1), LENGTH: 1400(345), HOP LIMIT: 46(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /74.125.130.95, PORT: 80(1), LENGTH: 1400(345), HOP LIMIT: 46(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
TCP PACKET. SOURCE IP: /106.10.193.45, PORT: 80(1), LENGTH: 0(345), HOP LIMIT: 46(0), SYN:0, ACK: 1, RST: 0, FIN: 1, NORMAL
UDP PACKET. SOURCE IP: /192.168.1.1, PORT: 53, LENGTH: 223(345), HOP LIMIT: 255(1), NORMAL
TCP PACKET. SOURCE IP: /173.194.36.6, PORT: 443(0), LENGTH: 1400(345), HOP LIMIT: 56(0), SYN:0, ACK: 1, RST: 0, FIN: 0, SYN Flood
UDP PACKET. SOURCE IP: /173.194.36.21, PORT: 443, LENGTH: 46(345), HOP LIMIT: 56(0), NORMAL
TCP PACKET. SOURCE IP: /173.194.36.1, PORT: 443(0), LENGTH: 30(345), HOP LIMIT: 56(0), SYN:0, ACK: 1, RST: 0, FIN: 0, NORMAL
```

Fig -6: Clustring

7. CONCLUSION

DoS attack are real treat as it not only affect the victim but also the authorize user of victim. So to detect such type of attack is necessary. In this paper, DoS attack detection system is designed by using layered real time detection approach on real time database generation.

ACKNOWLEDGEMENT

It is my privilege to acknowledge with deep sense of gratitude to my Project Guide for valuable suggestions and expert guidance throughout my course of study and timely help given to me in the completion of my project report.
I express my gratitude to Dr.V.B.Chowdhary, Principal, G.H.R.C.E.O.M and Prof. Tejaswini wakade(M.E. coordinator) for their kind help and co-operation.

REFERENCES

[1] G. Carl, G. Kesidis, R. R. Brooks, and S. Rai, "Denial-of-service attack-detection techniques," Internet Computing, IEEE, vol. 10

[2] Subramani rao Sridhar rao", "SANS InstituteInfoSec Reading Room., Denial of Service attacks and mitigation techniques: Real time implementation with detailed analysis",2011.

[3] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," ACM SIGCOMM Computer Communication Review, vol. 34, pp. 39-53, 2004..

[4] Aikaterini Mitrokotsa, Christos Douligeris, "Detecting Denial of Service Attacks Using Emergent Self-Organizing Maps", 2005 IEEE International Symposium on Signal Processing and Information Technology

[5] Kemal Bicakci, Bulent Tavli, "Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks", Computer Standards & Interfaces 31 (2009) 931–941, Elsevier

[6] Peng Ning, Ronghua Wang, Wenliang Du, "Containing Denial-of-Service Attacks in Broadcast Authentication in Sensor Networks", *MobiHoc'07,* September 9–14, 2007, ACM 978-1-59593-684-4/07/0009,Montr´eal, Qu´ebec, Canada.

[7] Kejie Lu, Dapeng Wu, Jieyan Fan, Sinisa Todorovic, Antonio Nucci, " Robust and efficient detection of DoS attacks for large-scale internet", Computer Networks 51 (2007) 5036–5056, 2007 Elsevier

[8] Tao peng, christopher leckie, kotagiri ramamohanarao, "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems", *ACM Comput. Surv. 39*, 1, Article 3 (April 2007)

[9] Mark Handley, Adam Greenhalgh, "Steps Towards a DoSresistant Internet Architecture", *SIGCOMM'04Workshops,* Aug. 30+Sept. 3, 2004, Portland, Oregon, USA. Copyright 2004 ACM 158113942X/04/0008

[10] Xiangjian He, Zhiyuan Tan, Aruna Jamdagni, Priyadarsi Nanda, Ren Ping Liu, Jiankun Hu, "Detection of Denial-of-Service Attacks Based on Computer Vision Techniques", IEEE TRANSACTIONS ON COMPUTERS

[11] Wanlei Zhou, Yang Xiang, Ke Li, "Low-Rate DoS Attacks Detection and Traceback by Using New Information Metrics", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 6, NO. 2, JUNE 2011

[12] Mieso K. Denko, "Detection and Prevention of Denial of Service (DoS) Attacks in Mobile Ad Hoc Networks using Reputation-Based Incentive Scheme", SYSTEMICS, CYBERNETICS AND INFORMATICS VOLUME 3 - NUMBER 4

[13] Phyu Thi Htun, Kyaw Thet Khaing, " Detection Model for Daniel-of-Service Attacks using Random Forest and k-Nearest Neighbors", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, No 5, May 2013

[14] Suresh Kumar, Anupama Chadha, "An Improved K-Means Clustering Algorithm: A Step Forward for Removal of Dependency on K", 2014 International Conference on Reliability, Optimization and Information Technology -ICROIT 2014, India, Feb 6-8 2014.

[15] M. Chinna Rao, K. Ramesh, G.Subbalakshmi "Decision Support in Heart Disease Prediction System using Naive Bayes", Indian Journal of Computer Science and Engineering (IJCSE), Vol. 2 No. 2 Apr-May 2011

[16] Rama Krishna Challa, Nidhi Srivastav, "Novel Intrusion Detection System integrating Layered Framework with Neural Network", *2013 3rd IEEE International Advance Computing Conference (IACC)*