Middle-East Journal of Scientific Research 23 (Sensing, Signal Processing and Security): 398-405, 2015
ISSN 1990-9233
© IDOSI Publications, 2015
DOI: 10.5829/idosi.mejsr.2015.23.ssps.211

DoS Attack Detection Based on Naive Bayes Classifier

[1]*V. Hema and* [2]*C. Emilin Shyni*

[1]AP/Department of CSE, India
[2]Professor/Department of CSE KCG College of Technology, India

Abstract: Interconnected systems, such as Web servers, database servers are now under threats from network attackers. Denial-of-service (DoS) attack is one such means which severely degrades the availability of a victim, which can be a host, a router, an entire network. They impose intensive computation tasks to the victim by flooding it with huge amount of useless packets. The victim is forced out of service from few minutes to several days. This causes serious damages to the services running on the victim. Therefore, effective detection of DOS attacks is essential for the protection of online services. A traffic classification scheme to improve classification performance when few training data are available is used. The traffic flows are described using the discretized statistical features and traffic flow information is extracted. A traffic classification method is proposed to aggregate the naïve bayes predictions of the traffic flows. Since classification scheme is based on the posterior conditional probabilities, it can identify attacks occurring in an uncertain situation The experimental results show that the proposed scheme can efficiently classify packets than existing traffic classification methods and achieved 92.34% accuracy.

Key words: DoS · Classifier · Naïve Bayes · Flodding

INTRODUCTION

This paper is to incorporate flow correlation analysis along with Naïve Bayesian classification process in order to determine the intruded packets in the network. To propose a mechanism for novel IDS named IDNB (Intrusion Detection using Naive Bayes) system specially designed for detecting intrusion packet where large data streams are arrived. In computing, a denial-of-service (DoS) or distributed denial-of-service (DDoS) attack is an attempt to make a machine or network resource unavailable to its intended users [1]. A common means of doing this is flooding the network by sending multiple requests to the server. Thereby, keeping the server busy for a long time and preventing legitimate users from getting the service.

The motive is to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet.

DOS attacks paralyze internet systems by overwhelming servers, network links and network devices with bogus traffic [2]. These attacks they do not target individual hosts, the attacks subdue the entire network. It becomes difficult to detect and handle the dos attack because a server would have received many number of such requests which may reduce its efficiency and it may take sometime to response to these requests. If all these requests were from malicious users then it takes some time for the legitimate users to receive the response. Therefore, suspending server access temporarily.

The system is able to detect DOS attack by intruding into the network and capturing the packets flowing in the network [3]. After capturing it extracts the header information or the whole packet information if necessary. The captured details undergo some pre-processing and are shown as traffic records.

A traffic record consists of the packet header information captured from the network with attribute and the corresponding value obtained after data pre-processing. A packet header information includes source ip address, destination port, captured packet length, hardware type, version, protocol, time to live, transmission type and many other details which are present in the packet.

A probabilistic classifier is a classifier that is able to predict, given a sample input, a probability distribution over a set of classes, rather than only predicting a class

Corresponding Author: V. Hema, AP/Department of CSE, India.

for the sample. Naive Bayes is a probabilistic classifier which is highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. NB classifier requires a small amount of training data to estimate the parameters of a classification model.

Training data is the data which is the base for building the model. It contains network traffic records captured at a particular time depicting different attribute values for which an attack can occur and the possibilities of a safe scenario [4]. It is from this training data that the Naïve Bayes classifier gains knowledge and based on this knowledge the classifier is able to predict a probabilistic value for the given scenario.

The usage of Naïve Bayes classifier seems to be the suitable solution in a network security scenario because of its predictability feature which is helpful in an uncertain world. It is just showing the possibility or probability value over the class labels and hence it clearly depicts the percent safety as well as the percent risk involved.

There are 3 primary issues that needs to be addressed while implementing the project.

Monitoring the Network: The network monitoring is the most difficult challenge to be performed because of the different types of request received. The important challenge is to categorize those requests and keep track of their source and destination ip address to ensure that it is not a spoofed address.

Statistical Analysis: All the necessary details from the packet need to be extracted and stored for statistical analysis. The complication lies with the different types of packets which is receive when the system is connected to a LAN. A detailed pie graph is shown with respect to the number of packets received of a particular type. This must be done for dynamic data where the main complications arise.

Classification Process: The difficulty with the classification process is that choosing appropriate training dataset. Since further classification of the traffic flow is based on the correctness, accuracy, possibilities of different scenarios of the training dataset, it must be handled with care.

Literature Survey: A new distributed approach to detecting DDoS [2] (distributed denial of services) flooding attacks at the traffic flow level. The new defense system is suitable for efficient implementation over the core networks operated by Internet service providers (ISP). At the early stage of a DDoS attack, some traffic fluctuations are detectable at Internet routers or at gateways of edge networks.A distributed change-point detection (DCD) architecture using change aggregation trees (CAT) has been devoloped. The idea is to detect abrupt traffic changes across multiple network domains at the earliest time. The system is built over attack-transit routers, which work together cooperatively. Each ISP domain has a CAT server to aggregate the flooding alerts reported by the routers. CAT domain servers collaborate among themselves to make the final decision.

A model of a real-time intrusion-detection expert system capable of detecting break-ins, penetrations and other forms of computer abuse is described [3]. The model is based on the hypothesis that security violations can be detected by monitoring a system's audit records for abnormal patterns of system usage. The model includes profiles for representing the behavior of subjects with respect to objects in terms of metrics and statistical models and rules for acquiring knowledge about this behavior from audit records and for detecting anomalous behavior. The model is independent of any particular system, application environment, system vulnerability, or type of intrusion, thereby providing a framework for a general-purpose intrusion-detection expert system.

The Internet and computer networks are exposed to an increasing number of security threats. With new types of attacks appearing continually, developing flexible and adaptive security oriented approaches is a severe challenge. Anomaly-based network intrusion detection techniques are a valuable technology to protect target systems and networks against malicious activities [5]. The System begins with a review of the most well-known anomaly-based intrusion detection techniques. Finally, an outline of the main challenges to be dealt with for the wide scale deployment of anomaly-based intrusion detectors, with special emphasis on assessment issues.

Related Work: Interconnected systems, such as Web servers, database servers, cloud computing servers etc, are now under threads from network attackers. As one of most common and aggressive means, Denial-of-Service (DoS) attacks cause serious impact on these computing systems. Existing system present a DoS attack detection system that uses Multivariate Correlation Analysis (MCA) [4] for accurate network traffic characterization by extracting the geometrical correlations between network traffic features. MCA-based DoS attack detection system employs the principle of anomaly-based detection in attack recognition. This makes the solution capable of detecting known and unknown DoS attacks effectively by learning the patterns of legitimate network traffic only. Furthermore, a triangle-area-based technique is proposed to enhance and to speed up the process of MCA.

This makes the solution capable of detecting known and unknown DoS attacks effectively by learning the patterns of legitimate network traffic only.

The existing system achieves a moderate overall detection rate. But there is some degradation in the attack detection. Analysis shows that the problem comes from the data used in the evaluation, where the basic features in the non-normalized original data are in different scales. The changes appearing in some other more important features with much smaller values can hardly take effect in distinguishing the DOS attack traffic from the legitimate traffic. The non-normalized original data contains zero values in some of the features and they confuse the MCA and make many new generated features equal to zeroes. The detection rate decreases when the network traffic increases showing high false positive rates and low detection rates.

Proposed Architecture: In proposed system, IDNB (Intrusion Detection by Naive Bayes) has been implemented. The main purpose of this implementation is to detect intrusion packets or data for increasing the performance of processing model. The proposed scheme is able to incorporate flow correlation information in to the classification process. IDNB (Intrusion Detection by Naive Bayes) demonstrates higher malicious behavior detection rates in certain circumstances while does not greatly affect the network performances. NB is one of the earliest classification methods applied in intrusion detection system which is an effective probabilistic classifier employing the Bayes' theorem with naive feature independence assumptions.

A probabilistic classifier is a classifier that is able to predict, given a sample input, a probability distribution over a set of classes, rather than only predicting a class for the sample. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers. NB classifier requires a small amount of training data to estimate the parameters of a classification model.

Detect intrusion packets data in client side when large complex data arrived. To minimize the effort of handling large complex data by using specialized tool used for securing network and checking available service. It provides security against hackers, malicious software, Denial of services. NB with feature discretization demonstrates not only significantly higher accuracy but also much faster classification speed. NB-based traffic classifier improves classification with a small set of training samples.

The system can be used to determine attacker packets in a Local Area Network (LAN) by capturing the packets flowing in the network. In order to capture the packets flowing in the network, the tool winpcap is used to analyze packets, transmit network packets, by-pass the protocol stack and monitor the network, network intrusion detection. The 'jpcap' distribution includes a tool for real time network traffic capture and analysis and an API for devoloping packet capture applications in java. The 'jpcap' network capture tool performs real-time decompostion and visualization of network traffic.

A statistical analysis is performed on the data analyzed from the packets flowing in the network captured by 'jpcap'. This analysis shows clear information regarding the types of packets flowing, cumulative count of packets of a specific type, size of these packets, total number of packets and graphical representation of network protocol ratio.

Apart from this a statistical representation is shown for types of packets flowing in a network and their count. Transport layer protocol ratio is also calculated and displayed in the form of a graph for the packets. Free memory available at any particular time can also be checked. A special feature about this is that it can perform the same calculation for static data as well as for dynamic data. Naïve Bayes with feature discretization demonstrates not only significantly higher accuracy but also much faster classification speed. Naïve Bayes based traffic classifier improves classification with a small set of training samples. Detect intrusion packets data in client side when large complex data arrived. The classification process proposed above is a machine learning algorithm which can classify intruded packets much more efficiently than the existing system since the problem scenario is an uncertain situation.

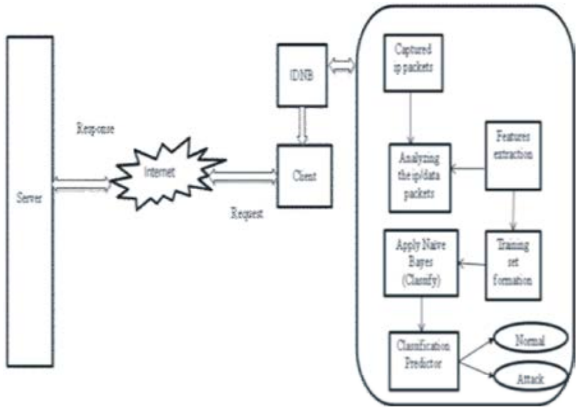


Fig. 1: Architecture Diagram

Response/Data Packets: The figure 1 is a system architecture which depicts the integration of each and every module into a single system. The entire system begins with the packet capturing operation followed by preprocessing to retrieve data. Later features are extracted for training set and test set data. Finally Naïve Bayes classification is applied to the test data set to classify the packets.

A server and a client are connected through a network. The client sends a request to the server and gets its service in response. A server may have many clients and it is capable of processing requests from multiple clients at a time. But this is limited depending on the capacity of the system. The IDNB system is implemented in between the client and the server. Its job is to check whether the packets a server is receiving is from normal user or not.

In order to do this it actually intrudes into the network to capture the packet flowing in the network. The winpcap and 'jpcap' are actually used for capturing packets. winpcap which also helps us to analyze packets, transmit network packets, by-pass the protocol stack, monitor the network, network intrusion detection. The 'jpcap' distribution includes a tool for real time network traffic capture and analysis and an API for devoloping packet capture applications in java. The 'jpcap' network capture tool performs real-time decompostion and visualization of network traffic.

The next step is to perform a pre-processing that is retrieve header data from the packets which needs to be stored and displayed. Those values which are not available are represented using the Not Available value.

It is from these pre-processed data suitable features are extracted for performing the classification.Feature extraction involves choosing the parameters or the header data which should be chosen to classify packets. The extracted features must be discrete to use it classification. So, if it is not discretized they should be discretized before proceeding to the next step.

Now a decision is made on how the training data should be and what all a training data should have in it. The training data plays an important role in the classification process because it is based on the accuracy of the training data and different possibilities of an attack and attackless situations in the training data, a classifier would be able to perform effectively in the uncertainity world with the help of the posterior conditional probability formula.

Pre-processing should also be done for a test data set before supplying the dataset to the classifier. Then Naïve Bayes classification is applied to the test data set which reveals the final output of the system stating whether the packet is an attacker packet or a normal packet.

Pre-Processing: A data set (or dataset) is a collection of data, usually presented in tabular form. Each column represents a particular variable. It lists values for each of the variables, such as source ip address, destination port, captured packet length and other header details of an IP packet. Each value is known as a datum. The data set may comprise data for one or more members, corresponding to the number of rows. The values may be numbers, such as real numbers or integers, but may also be nominal data (*i.e.*, not consisting of numerical values). More generally, values may be of any of the kinds described as a level of measurement. There may also be "missing values", which need to be indicated as Not Available.

Data pre-processing is an important step in the data mining process. Data-gathering methods are often loosely controlled, resulting in out-of-range values (e.g., Income: -100), impossible data combinations (e.g., Sex: Male, Pregnant: Yes), missing values, etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running an analysis.

If there is much irrelevant, redundant or noisy and unreliable data present, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. Data pre-processing includes cleaning, normalization, and selection. The output of data pre-processing is the traffic records.

Data cleansing, data cleaning or data scrubbing is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database. Used mainly in databases, the term refers to identifying incomplete, incorrect, inaccurate, irrelevant, etc. parts of the data and then replacing, modifying, or deleting this dirty data or coarse data.

After cleansing, a data set will be consistent with other similar data sets in the system. The inconsistencies detected or removed may have been originally caused by user entry errors, by corruption in transmission or storage, or by different data dictionary definitions of similar entities in different stores.

Data normalization is the process of reducing data to its canonical form. For instance, Database normalization is the process of organizing the fields and tables to minimize redundancy and dependency. In the field of software security, a common vulnerability is unchecked malicious input.

Feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features for use in model construction. The central assumption when using a feature selection technique is that the data contains many redundant or irrelevant features. Redundant features are those which provide no more information than the currently selected features and irrelevant features provide no useful information in any context.

Feature selection techniques provide two main benefits they are improved model interpretability, shorter training times. Feature selection is also useful as part of the data analysis process, as it shows which features are important for prediction and how these features are related.

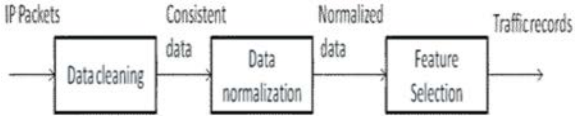
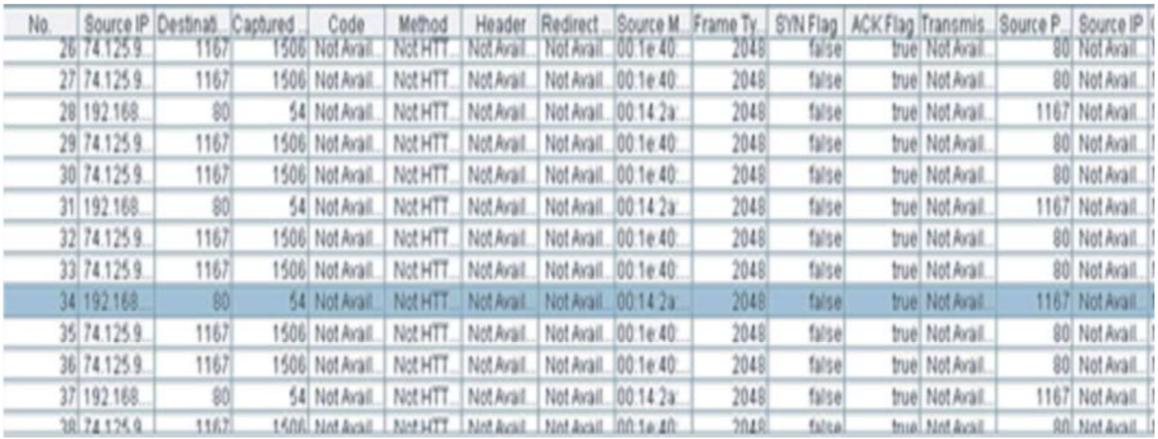


Fig. 2: Pre-Processing

Packet Discretization: It is based on a flow-level traffic classification. The system captures IP packets crossing a target network and constructs traffic flows by checking the headers of IP packets. It is flow-level traffic classification. A flow consists of successive IP packets with the same 5-tuple: source IP, source port, destination IP, destination port and transport layer protocol. It uses heuristic way to determine the correlated flows and model them. If the flows observed in a certain period of time share the same destination IP, destination port and transport layer protocol, they are determined as correlated flows and form a BoF(Bag of Flow). For the classification purpose, a set of flow statistical features are extracted and discretized to represent traffic flows.

A discrete data attribute can be seen as a function whose range is a finite set, while a continuous data attribute as a function whose range is an infinite totally ordered set, usually an interval. To discretized a continuous data attribute means to find a partition of the range of that attribute into a finite number of intervals. The discretization process consists of two steps. First, the number of discrete intervals needs to be chosen. Second, the cut points must be determined for performing packet discretzation.

Classification Process: Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes theorem with the "naive" assumption of independence between every pair of features. It is used to classify



No.	Source IP	Destinati...	Captured...	Code	Method	Header	Redirect...	Source M...	Frame Ty...	SYN Flag	ACK Flag	Transmis...	Source P...	Source IP
26	74.125.9...	1167	1506	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:1e:40:...	2048	false	true	Not Avail...	80	Not Avail...
27	74.125.9...	1167	1506	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:1e:40:...	2048	false	true	Not Avail...	80	Not Avail...
28	192.168...	80	54	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:14:2a:...	2048	false	true	Not Avail...	1167	Not Avail...
29	74.125.9...	1167	1506	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:1e:40:...	2048	false	true	Not Avail...	80	Not Avail...
30	74.125.9...	1167	1506	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:1e:40:...	2048	false	true	Not Avail...	80	Not Avail...
31	192.168...	80	54	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:14:2a:...	2048	false	true	Not Avail...	1167	Not Avail...
32	74.125.9...	1167	1506	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:1e:40:...	2048	false	true	Not Avail...	80	Not Avail...
33	74.125.9...	1167	1506	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:1e:40:...	2048	false	true	Not Avail...	80	Not Avail...
34	192.168...	80	54	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:14:2a:...	2048	false	true	Not Avail...	1167	Not Avail...
35	74.125.9...	1167	1506	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:1e:40:...	2048	false	true	Not Avail...	80	Not Avail...
36	74.125.9...	1167	1506	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:1e:40:...	2048	false	true	Not Avail...	80	Not Avail...
37	192.168...	80	54	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:14:2a:...	2048	false	true	Not Avail...	1167	Not Avail...
38	74.125.9...	1167	1506	Not Avail...	Not HTT...	Not Avail...	Not Avail...	00:1e:40:...	2048	false	true	Not Avail...	80	Not Avail...

Fig. 3: Discretized Traffic Records

the result for any given data set (test data) based on the dataset provided with their accurate results (training data).

Bayes theorem is stated mathematically as the following equation

$$P(A|B)=\frac{P(B|A)\,P(A)}{P(B)}$$

where,
A and *B* are events.

P(*A*)and P(*B*)are the probabilities of *A* and *B* independent of each other.

P(*A*|*B*), a conditional probability, is the probability of *A* given that *B* is true.

P(*B*|*A*), is the probability of *B* given that *A* is true.

Supervised learning is the machine learning task of inferring a function from labeled training data. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances.

Single Naïve Bayes Predictor: Naïve Bayes algorithm to produce a set of posterior probabilities as predictions for each testing flow. It is different to the conventional NB classifier which directly assigns a testing flow to a class with the maximum posterior probability. Considering correlated flows, the predictions of multiple flows will be aggregated to make a final prediction.

The posterior probability of a random event or an uncertain proposition is the conditional probability that is assigned after the relevant evidence or background is taken into account. The posterior probability distribution is the probability distribution of an unknown quantity, treated as a random variable, conditional on the evidence obtained from an experiment or survey. "Posterior", in this context, means after taking into account the relevant evidence related to the particular case being examined.

In probability theory, a conditional probability measures the probability of an event given that (by assumption, presumption, assertion or evidence) another event has occurred.

A random event can be defined as an event whose outcome cannot be predicted in advance.

Aggregated Predictor: A number of combination methods can be derived from the Bayesian decision theory which can be used for aggregated predictor.

The basic formula used for predicting the output is;

P(y | x1,…x_n) 8 p(y)?n1 p(xi | y)

where,
x_n=>Traffic Records{src ip, dest port, cap len, method, src mac, frame type, dest mac, SYN flag, ACK flag}
y=>attack{yes/no}
For eg,
P(no|10.2.0.15,80,299,GET,08:00:27:9d:bf:60,2048,52:54:00:12:35:02,false,true)

System Implementation

Packet Capturing: This part captures the packets flowing in a network, here, it is a Local Area Network and the required details are stored. The packet capturing process can be modified by choosing the options such as header only full packet options. After the packets are captured count of the packets are displayed and immediately it is predicted whether the packet is an attack or not.

```
public Capture() throws java.io.IOException
{if((System.currentTimeMillis()-timeo)>1000){
Test1 pc = new Test1();
Packet packe = pc.pack;
{packet = pc.getPackets();
tcpcount += pc.tcpp;
udpcount += pc.udpp;
arpcount += pc.arpp;
System.out.println("tcpcount:"+tcpcount);
System.out.println("udpcount:"+udpcount);
System.out.println("arpcount:"+arpcount);}
```

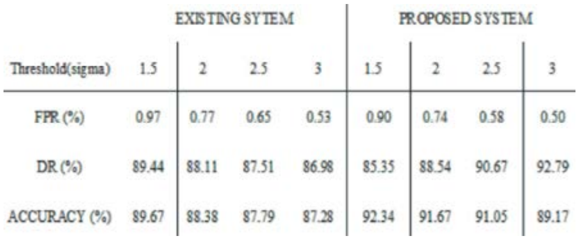
Statistical Analysis: The statistical analysis shows the number of packets received of a particular type and the ratio is displayed in a pie chart format. It also displays a pie chart depicting the cumulative network protocol ratio and transport protocol ratio. The overall information shows the total number of packets received, cumulative size of packets and their average.

```
JDCumlativeStatFrame(Vector packets,JDStatisticsTaker
staker){super(staker.getName());
this.staker=staker;
staker.analyze(packets);
getContentPane().setLayout(new BoxLayout (getContent
Pane(),BoxLayout.Y_AXIS));
model=new TableModel();
table=new JTable(model);
table.setSelectionMode(ListSelectionModel.SINGLE_S
ELECTION);
JTableHeader header = table.getTableHeader();
Dimension dim = header.getPreferredSize();
dim.height=20;
header.setPreferredSize(dim);
JScrollPane tablePane=new JScrollPane(table);
dim=table.getMinimumSize();
dim.height+=25;
tablePane.setPreferredSize(dim);
```

Attack Detection: The received packet information is used to analyze whether they are attacker packets or normal packets. The Naïve Bayes algorithm actually takes a training data set which is being used to evaluvate the received packets. It requires only few training data to perform this but they should be able to depict different attacked and safe scenario.

```
panel = new JPanel();
panel1 = new JPanel();
tcp = new JLabel(" TCP");
//Initialising new components
// start.addActionListener(this);
// start.addActionListener(this);
//Add the swing components into the panel
PieGraph piegraph = new PieGraph(labels,values);
//Creating pie graph objects
//Add the componenets into the panel
getContentPane().setLayout(new BorderLayout());
//Setting layout for each panel
setSize(800,600);
setVisible(true);
```

Result and Discussion: The proposed system is able to improve the detection rate and reduce the occurrence of false positive alarms in the system for detecting denial of service attack using Naïve Bayes classification. The system can detect DOS attack occurring in the Local Area Network(LAN). In the existing system false positive rate decreases with increase in threshold and the accuracy also decreases with the increase in the threshold. The proposed system is able to produce minor improvement in these detection rates compared to the existing system. Accuracy in detecting the DOS attacks are also increased. But the detection rate and accuracy decreases with increase in traffic.



	EXISTING SYTEM				PROPOSED SYSTEM			
Threshold(sigma)	1.5	2	2.5	3	1.5	2	2.5	3
FPR (%)	0.97	0.77	0.65	0.53	0.90	0.74	0.58	0.50
DR (%)	89.44	88.11	87.51	86.98	85.35	88.54	90.67	92.79
ACCURACY (%)	89.67	88.38	87.79	87.28	92.34	91.67	91.05	89.17

Fig. 4: Comparison of Detection Rate and False Positive Rates Achieving by the Proposed System and Existing System

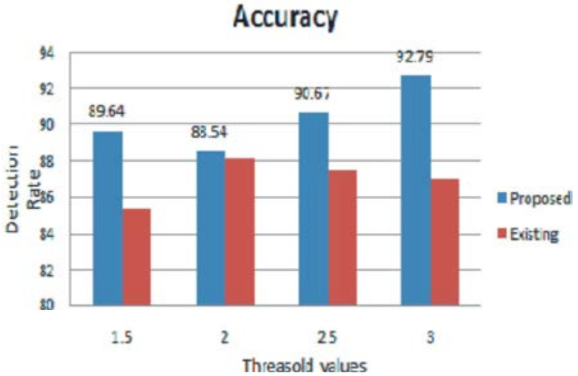


Fig. 5: Comparison of Detection Rate and False Positive Rates Achieving by the Proposed System and existing system

This graph shows that Naïve bayes classification provides more accuracy than the existing system.

CONCLUSION AND FUTURE WORK

Packet analysis has been shown as an approach of generating packet filters that combine most of the desired properties in terms of processing speed, memory consumption, flexibility and simplicity in specifying protocol formats and filtering rules, effective filter composition and low run-time overhead for safety enforcement. The development of the filter generator and the test results support the viability of our claims. It aims at emitting fast and efficient filters while preserving all the relevant safety properties, both in terms of memory access correctness and termination.

The system detects attacks in a Local Area Network(LAN) and can be further expanded to implement in a Wide Area Network(WAN). The other feature that can be added in future is to include pattern recognition algorithms for intrusion detection purpose since the tool is able to pre-process the network connection data. Further enhancements are performing some preliminary analysis for network intrusion detection such as clustering similar network connections according to similar patterns e.g., payload or network protocols.

REFERENCES

1. Chen, Y. and Kai Hwang, 2007. "Collaborative Detection of DDoS Attacks over Multiple Network Domains," IEEE Trans., Parallel and Distributed System, pp: 18.
2. Denning, D.E., 1987. "An Intrusion-Detection Model," IEEE Trans., Software Eng., vol. TSE-13, 2: 222-232.
3. Garca-Teodoro, P., J. Daz-Verdejo, G. Maci-Fernndez and E. Vzquez, 2009. "Anomaly -Based Network Intrusion Detection: Techniques, Systems and Challenges," Computers and Security, 28: 18-28.
4. Jin, S., D.S. Yeung and X. Wang, 2007. "Network Intrusion Detection in Covariance Feature Space," Pattern Recognition, 40: 2185-2197.
5. Paxson, V., 1999. "Bro: A System for Detecting Network Intruders in Real Time, " Computer Networks, 31: 2435-2463.