International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 5 1136– 1142

Detecting and Classifying Attacks using Artificial Neural Network

Reema Jaggi
Computer Engineering Department
PES Modern College of Engineering
Pune, India
reema29j@gmail.com

Jidnyasa Sangade
Computer Engineering Department
PES Modern College of Engineering
Pune, India
jidnyasak.sangade@gmail.com

Abstract - As computers are becoming increasingly used by businesses; security issues have posed a big problem within organizations. Firewalls, anti-virus software, password control are amongst the common steps people take towards protecting their systems. However, these preventive measures are not perfect. Firewalls are vulnerable; they maybe improperly configured or may not be able to prevent new types of attacks. Ant-virus software works only if the virus is known to the public. Passwords can be stolen and therefore, systems can be easily hacked into. Hackers can change the system on initial access and manipulate it so that their future access will not be detected. In these situations, Intrusion Detection Systems (IDS) come into play. This paper presents a new approach of IDS based on neural network. We have use Multi-Layer Perceptron based on Back Propagation. It is capable of detecting denial of service, probe attacks, user to root and root to local attack. Our proposed system not only detects attacks but also classify them in 6 categories with the accuracy of approximately 90.78%.

Keywords- *Intrusion Detection System, Artificial Neural Network, Multi-Layer Perceptron algorithm, Apriori algorithm*

*****

I. INTRODUCTION

There is increase in dependency of companies and government agencies have on their computer networks the importance of protecting these systems from attack is critical [3]. During recent years, number of attacks on networks has dramatically increased and consequently interest in network intrusion detection has increased among the researchers. The rapid development and expansion of World Wide Web and local network systems have changed the computing world in the last decade. The highly connected computing world has also equipped the intruders and hackers with new facilities for their destructive purposes. The costs of temporary or permanent damages caused by unauthorized access of the intruders to computer systems have urged different organizations to increasingly implement various systems to monitor data flow in their networks. These systems are generally referred to as Intrusion Detection Systems (IDSs) [7].

Different structures of MLP are examined to find a minimal architecture that is reasonably capable of classification of network connection records. The results show that even an MLP with a single layer of hidden neurons can generate satisfactory classification results. Because the generalization capability of the ids is critically important, the training procedure of the neural networks is carried out using a validation method that increases the generalization capability of the final neural network. Network intrusion detection system based on artificial neural network not only detects normal or attack connection but also classify the attacks into attack types. For training Back Propagation algorithm is used. Back propagation is proposed by **David E. Rumelhart Geoffrey E. Hinton Ronald J. Williams**. Back propagation with 2 hidden is implemented which not only divide the data in to normal or attack but also provide solution to more general problem by classifying the attacks in to attack types. Batch update of weights provides advantageous to weight correction term and it is relatively simple to implement and we can easily reduce the computing by choosing small weights in the beginning.

II. OVERVIEW

A. Intrusion Detection System

An IDS is a security monitoring system that will gather and analyse data from various areas within a system or network to identify/detect possible intrusions and/or misuse. An intrusion detection system (IDS) can be a key component of security incident response within organizations. Traditionally, intrusion detection research has focused on improving the accuracy of IDSs, but recent work has recognized the need to support the security practitioners who receive the IDS alarms and investigate suspected incidents [8]. Intrusion Detection Systems perform a wide array of functions, which include:

1) Monitoring and analyzing user and system activities,
2) Analyzing system configurations and vulnerabilities,
3) Ability to recognize patterns of typical attacks,
4) Analysis of abnormal activity patterns, and
5) Tracking user policy violations.

The system can be a key asset in pinpointing where attacks are coming from and when they are being made. It will also indicate the primary targets of the attack and the types of

attacks being utilized. While the complexities of host computers already made intrusion detection a difficult attempt, the increasing in spreading distributed network-based systems and insecure networks such as the Internet has greatly increased the need for intrusion detection [5], [6].

B. Types of IDS

There are several ways to categorize the types of IDS that are available. Based on the kind of activities, transactions, traffics or systems they monitor, IDS can be categorized into Network based (NIDS), Host based (HIDS) and Application based (APIDS). Based on differing approach to event analysis, IDS can be distinguished between Signature based (SIDS) and Anomaly based. Each type of IDS has its own strengths and weaknesses.

1) Signature Based IDS: Signature based IDS are based on looking for known patterns of detrimental activity. It monitors traffic and seeks a pattern or a signature match. Signature based IDS are very accurate. The systems are fast since they are only doing a comparison between what they are seeing and a predetermined rule. If someone develops a new attack, there will be no protection. They are only as strong as there rule set [9].

2) Anomaly Based IDS: Anomaly based IDS are based on tracking unknown unique behavior pattern of detrimental activity. It helps to reduce the limitations problem. It conducts a thorough screening of what comes through. The false positive detection rate is too much because Behavior based IDSs monitor a system based on their behavior patterns. It is painstaking slow to do an exhaustive monitoring, uses up a lot or resource [9].

3) Network based IDS: NIDS monitors the incoming and outgoing network traffic to identify intruders. They access the network through hubs or network taps [5]. They look for suspicious patterns by reading incoming network packets [6]. In addition, they also scan outgoing traffic for the possibility of an internal intruder. Due to the fact that both incoming and outgoing traffic is monitored and that the NIDS are placed in points within the network, it may slowdown the overall speed of the network.

4) Host based IDS: HIDS monitors incoming and outgoing activity on a particular system in the network. Specifically, it monitors the dynamic behaviour and state of the computer system. The administrator will be notified once an intrusion has been detected. An NIDS is usually used alongside a HIDS to identify any activities that HIDS overlooked [10].

5) Application protocol based IDS: APIDS monitors activity in the specific protocols used in the computer system. It looks for protocols and enforces the correct use in the systems [10].

C. Artificial Neural Network

An artificial neural network is composed of many neurons that are linked together according to specific network architecture. The objective of the neural network is to transform the inputs into meaningful outputs. The result is determined by the characteristics of the nodes and the weights associated with the interconnections among neurons. By modifying the connections between the nodes the network is able to adapt to the desired outputs [4]. It resembles the brain in two respects:

1) Knowledge is acquired by the network from its environment through a learning process.
2) Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

The procedure used to perform the learning process is called a learning algorithm. Each neuron is a simple processing unit which receives some weighted data, sums them with a bias and calculates an output to be passed on. The function that the neuron uses to calculate the output is called the activation function. The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network. The most common architecture is the multilayer perceptron (MLP). These networks are a feed forward network where the neurons are structured in one or more hidden layers. Each perceptron in one layer is connected to every perceptron on the next layer; hence information is constantly "fed forward" from one layer to the next. The network learns about the input through an interactive process of adjusting the weights and the bias. This process is called supervised learning and the algorithm used is the learning algorithm.

One of the most common is the error back propagation algorithm. This algorithm is based on the error-correction learning rule, based on gradient descent in the error surface. Basically, a set of cases, with the corresponding targets, is given to the network. The input data is entered into the network via the input layer and is processed through the layers - forward pass. Then, the output is compared to the expected output (the targets) for that particular input. This results in an error value. This error value is back propagated thought the network, against the direction of the weights. The weights and bias are adjusted to make the actual

 ISSN: 2321-8169
1136– 1142

response of the network move closer to the desired response in a statistical sense.

D. MLP Algorithm

A multilayer perceptron (MLP) is a feed-forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. A MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique called back-propagation for training the network.

Steps:

1. Initialize weights of every neuron with random values.
2. Forward pass: It consists in the propagation of the input signals to the output. Also, the calculation of the error made is done, comparing the obtained output with the desired output.
3. Apply input values to the neural network in order to calculate the output (i.e. actual output).
4. Compare the actual output with the desired output.
5. Backward pass: This procedure starts in the output layer, propagating the error signals to the input layer, and calculating recursively the local gradients for each neuron. The output error is used to alter weights on the output units. Then the error at the hidden nodes is calculated (by back-propagating the error at the output units through the weights), and the weights on the hidden nodes altered using these values.
6. Propagate backwards from output to input layers for corrections.
7. Compute weight deltas using equation

$$\Delta W_{ji}(n+1) = \eta o_i \sigma_j + \alpha \Delta W_{ji}(n)$$

where,

η is the learning rate which represent gradient descent step width.

α is the momentum which tell the amount of weight change of jth unit calculated by using the equation

where,

$$\sigma_j = \begin{cases} (f'net_j) + c)(desired\text{-}actual) & \text{if j is output layer} \\ (f'(net_j) + c)\Sigma_k \sigma_k W_{jk} & \text{if j is hidden layer} \end{cases}$$

c is the at spot elimination value

$f'(net_j)$ is the derivative of the activation function

8. Repeat from step 2 to step 7 to cover all patterns.
9. Update weight values of the neural network by adding weight deltas [11].

E. Apriori Algorithm

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database. Apriori is designed to operate on databases containing transactions. Each transaction is seen as a set of items (an itemset). Given a threshold C, the Apriori algorithm identifies the item sets which are subsets of at least C transactions in the database.
Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.
Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length K from item sets of length K-1. Then it prunes the candidates which have an infrequent sub pattern.

III. DATASET

The dataset Network Intrusion Detection was chosen from The UCI (University of California, Irvine) KDD (Knowledge Discovery in Databases) Archive. In CUP KDD dataset, each event (connection) is described with 41 features. 22 of these features describe the connection itself and 19 of them describe the properties of connections to the same host. The dataset also contains 300,000 instances and 38 different classes.

A. Categories of Attacks

All attacks fall into 4 main categories: Denial of Service (DOS), User to Root (U2R), Remote to Local (R2L) and Probe [7]. In this study six types of attacks are considered: Smurf, Teardrop, Satan, Guest, Warezclient and Buffer overflow. These six attack types were selected from four different attack categories denial of service, probing, user to root and remote to user.

1) *Denial of Service* (*DOS*): In computing, a denial-of-service attack is an attempt to make a machine or network resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of efforts to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet. Denial-of-service attacks can essentially disable your computer or your network.

Smurf and Teardrop belong to this category.

- *Smurf*: It is a distributed denial-of-service attack in which large numbers of Internet Control Message Protocol (ICMP) packets with the intended victim's spoofed source IP are broadcast to a computer network using an IP Broadcast address. Most devices on a network will, by default, respond to this by sending a reply to the source IP address. If the number of machines on the network that receive and respond to these packets is very large, the victim's computer will be flooded with traffic. This can slow down the victim's computer to the point where it becomes impossible to work on.

- *Teardrop*: It is a program that sends IP fragments to a machine connected to the Internet or a network. Teardrop exploits an overlapping IP fragment bug present in Windows 95, Windows NT and Windows 3.1 machines. The bug causes the TCP/IP fragmentation re-assembly code to improperly handle overlapping IP fragments. This attack has not been shown to cause any significant damage to systems, and a simple reboot is the preferred remedy. It should be noted, though, that while this attack is considered to be non-destructive, it could cause problems if there is unsaved data in open applications at the time that the machine is attacked. The primary problem with this is a loss of data.

2) *User to Root* (U2R): These attacks are exploitations in which the hacker starts off on the system with a normal user *account* and attempts to abuse vulnerabilities in the system in order to gain super user privileges e.g. perl, xterm [2].

- *Buffer Overflow*: These types of attacks are designed to trigger arbitrary code execution by a program by sending it more data than it is supposed to receive. Programs that accept parameterized input data temporarily store them in a region of memory called a buffer). But some read functions, such as strcpy() functions from the C language, cannot manage this type of overflow and cause the application to crash, which can lead to arbitrary code execution and open access to the system.

3) *Remote to Local* (R2L): A remote to user attack is an attack in which a user sends packets to a machine over the internet, which s/he does not have access to in order to expose the machines vulnerabilities and exploit privileges which a local user would have on the computer e.g. xlock, guest, xnsnoop, phf, sendmail dictionary [2].

- *Guest*: The Guest attack is a variant of the Dictionary attack that the dictionary attack is a Remote to Local User attack in which an attacker tries to gain access to some machine by making repeated guesses at possible user names and passwords. On badly configured systems, guest accounts are often left with no password or with an easy to guess password. Because most operating systems ship with the guest account activated by default, this is one of the first and simplest vulnerabilities an attacker will attempt to exploit [1].

- *Warezclient*: Warezclient attack can be launched by any legal user during an FTP connection after warezmaster attack has been executed. During warezclient attack users download the illegal “warez” software that was posted earlier through a successful warezmaster attack [1].

4) *Probe*: Probing is an attack in which the hacker scans a machine or a networking device in order to determine *weaknesses* or vulnerabilities that may later be exploited so as to compromise the system. This technique is commonly used in data mining e.g. saint, portsweep, mscan, nmap etc [2].

- *Satan*: It is a publicly available tool that probes a network for security vulnerabilities and misconfigurations. It is sometimes used by administrators and often used by attackers to search for vulnerabilities on a network. Information provided by SATAN could be useful to an attacker in performing an attack. The shareware version of SATAN was widely distributed on the Internet and is still used within the Internet community at large

 ISSN: 2321-8169
1136– 1142

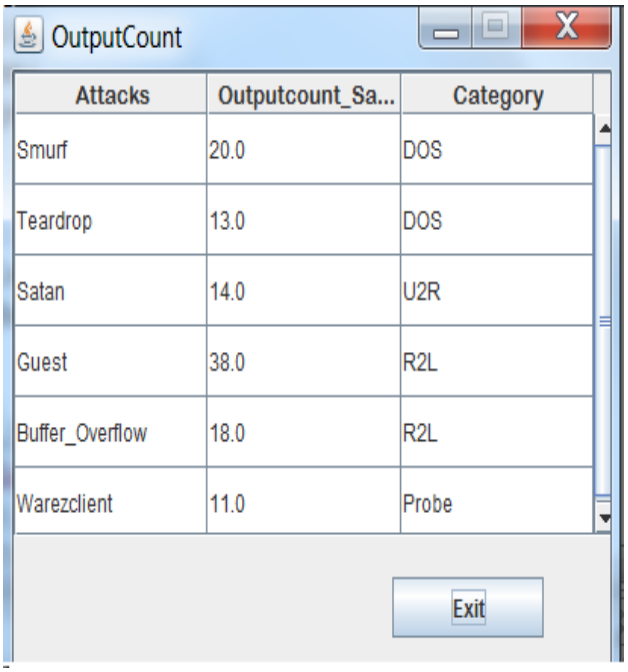

Fig1. Names, Types and Numbers of Attacks in 10% of Dataset

IV. ARCHITECTURE OF PROPOSED SYSTEM

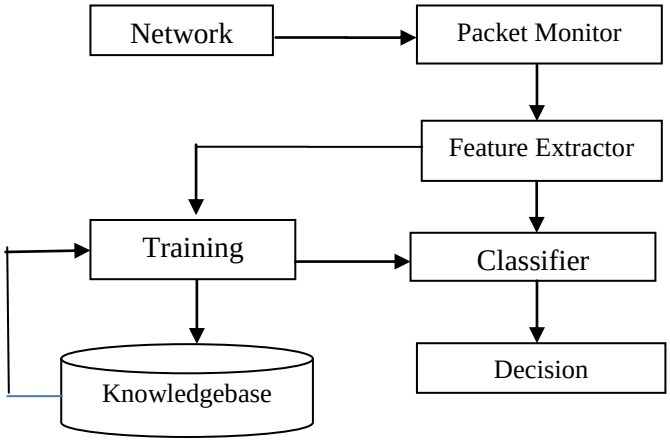

Fig2. System Overview

A. *Packet Monitor*: This module monitors network stream real time and capture packets to serve for the data source of the NIDS.

B. *The Feature Extractor*: This module extracts feature vector from the network packets (connection records) and submits the feature vector to the classifier module. The features extracted from the network stream are a feature vector which serves for the description of the packet. A complete description of all 41 features is available. The features like flag, count, src_bytes, dst_bytes, etc.

C. *The Classifier*: This module provides the processed data for training to the neural network and draws the conclusion about the detection of intrusion. It is used to differentiate "Good Packets" from "Bad Packets" in the network stream. In this sense, "Intrusion Detection" can be understood as "Pattern Recognition" to some extent.

D. *The Training Layer*: It provides training to the neural network. The algorithm used is MLP back-propagation from WEKA tool box.

E. *The Decision Layer*: It provides the detection rate and false alarm rate of the network intrusion detection system.

1) *False alarm rate*: In this situation, an attack occurs against your network and your IDS fails to alarm even though it is designed to detect such an attack. Your IDS should almost never generate false negatives. In fact, it is preferable for your IDS to actually generate more false positives rather than generating any false negatives. False Alarm Rate defined as the number of 'normal' patterns classified as attacks (False Positive) divided by the total number of 'normal' patterns

2) *Detection rate*: The detection rate is defined as the number of intrusion instances detected by the system (True Positive) divided by the total number of intrusion instances present in the test set.

F. *The Knowledge Base Layer*: It serves for the training samples of the classifier phase. As you know, the artificial neural networks can work effectively only when it has been trained correctly and sufficiently. The intrusion samples can be perfected under user participation, so the capability of the detection can improve continually.

V. IMPLEMENTATION

Proposed system divided into two parts in first part we are detecting and classifying attacks using MLP back-propagation algorithm. And in the second part real time detection of icmp attacks .By assuming icmp packets using modified apriori algorithm rule is generated and using Snort IDS we can detect the attack and if the attack detected in client machine in the response server machine kills the process.

The first part consists of 3 modules namely:

 ISSN: 2321-8169
1136– 1142

A. *Data Labeling*

We have to select two file on is text file and another is in arff format. Text file is used to count the attacks for comparison with the MLP output. Arff file format is provided as input to the MLP and after selecting file. Data labeling is performed and normal is represented as +1 and attack is -1. And we can easily see the output of data labeling.

B. *MLP Algorithm and Attack Classification*

This module is constructed using 2 class, MLP input and MLP output. We get the input count of each attack along with its category and the output which is the count and category is given in the table format by clicking button labeled as MLP input. We will run MLP algorithm which provide in the result count of each attack and its category which is the output of MLP actually output of MLP is confusion matrix in which its diagonal provides the count of each attack. For showing it in the table we just fetch the diagonal values. And display it by matching it with attribute label.

C. *Calculating Detection Rate, False Alarm Rate, Apriori Algorithm*

The detection rate of the proposed system is calculated along with the false positive rate and false negative rate of each attack. A graph is plot for detection rate and false alarm rate.

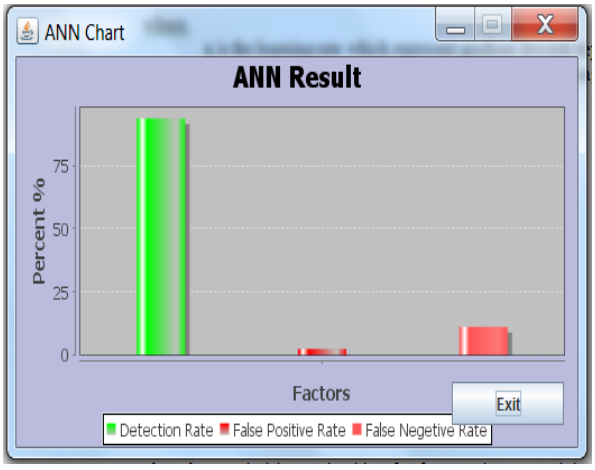

Fig3. Detection Rate

The second part consists of 1 module:

A. *Real Time Detection using Snort Tool*

This module deals with real time detection using Snort IDS we are detecting only ICMP attacks because here we are assuming incoming packets are only ICMP packets and using modified signature apriori algorithm rule are generated for ICMP attacks and stored in the IDS which helps to detect the attacks.

1) Snort Tool

Snort is the leading open source Network Intrusion Detection System and is a valuable addition to the security framework at any site. Even if you are employing lots of preventative measures, such as firewalling, patching, etc., a detection system can give you an assurance that your defences truly are effective, or if not, will give you valuable information about what you need to improve. Fortunately, there is a good set of snort packages for Debian which takes a lot of the tedious work out of building a useful Network Intrusion Detection System. Before we start on installation, we should review a few details about the networking stack that you're going to need to make sense of the alerts snort will generate. Impatient readers and those who are familiar with the TCP/IP suite of protocols may do now skip to the bit that says standalone snort.

VI. CONCLUSIONS

In this paper, we present and implemented an Intrusion Detection System by applying genetic algorithm to efficiently detect various types of network intrusions. To implement and measure the performance of our system we used the standard KDD99 benchmark dataset and obtained reasonable detection rate. The system detects as well as classifies them into category. Our proposed system not only detects attacks but also classify them in 6 categories with the accuracy of approximately 90.78%. In future we may detect UDP and TCP attacks. In future we can classify more attack type and proper action can be taken against attacks types detected.

ACKNOWLEDGMENT

We would like to express our deepest thanks to few people who have helped us a lot in designing and working with the project, without whom it would have been impossible for us to come up with such a good solution for automating the manual task of operating system installation. We would explicitly like to name our guide Mrs. Pallavi Baviskar for her. A special thanks to all my friends for being so supportive.

REFERENCES

[1] Mohammad Reza Norouzian and Sobhan Merati,"Classifying Attacks in a Network Intrusion Detection System Based onArtificial Neural Networks," Information Technology Department, Institute for

Advanced Studies in Basic Sciences, Zanjan, Iran, Feb 2011.

[2] Mohammad Sazzadul Hoque, Md. Abdul Mukit2 and Md. Abu Naser Bikas, "An Implementation Of Intrusion Detection System Using Genetic Algorithm", Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet, Bangladesh, Vol.4, No.2, March 2012.

[3] James Cannady, " Artificial Neural Network," Proceedings of the 1998 National Information System Security Conference ,Arlington VA,1998.

[4] Mukherjee, B., Heberlein, L.T., Levitt, K.N, "Network Intrusion Detection". IEEE Network. pp. 28-42, 1994.

[5] Kabiri P, Ghorbani A A. "Research in intrusion detection andresponse - a survey". International Journal of Network Security, 2005.

[6] Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, "Cost-based Modeling and Evaluation for Data Mining With Application to Fraud and Intrusion Detection: Results from the JAM Project," This research is supported in part by grants from DARPA (F30602-96-1-0311) and NSF (IRI- 96-32225 and CDA-96-25374), 27 August 1998.

[7] Mehdi Moradi and Mohammad Zulkernine, "A Neural Network Based System for Intrusion Detection and Classification of Attacks".

[8] Rodrigo Werlinger, Kirstie Hawkey, Kasia Muldner, Pooya Jaferian,Konstantin Beznosov," The Challenges of Using an Intrusion Detection System: Is It Worth the Effort?" University of British Columbia, Vancouver, Canada.

[9] Stephen Loftus and Kent Ho, "Signature Based and Anomaly Based Network Intrusion Detection".

[10] Yanny Liu, "An Intorduction To Intrusion Detcetion System", 5 July 2009.

[11] Daniel del and Hoyo Rodriguez, "Optimization of Back-propagation Learning Algorithm on MLP Networks", 2 Nov 2012.