A Machine Learning Approach to Network Intrusion Detection

Keerthi M Venugopal[1] Ayushi Shekhar[2] Bhavya N R[3] Mrs. S. Kuzhalvaimozhi[4]
[4]Assistant Professor
[1,2,3,4]Department of Information Science and Engineering
[1,2,3,4]The National Institute of Engineering, Mysore-08

Abstract— Network Intrusion (forcefully entering a network) Detection Systems are now a critical means to provide security against malicious activities. In network intrusion detection research, one of the popular strategies for finding attacks is monitoring a network's activity for anomalies which are deviations from normal profiles previously learned from benign traffic, identified typically using tools borrowed from the machine learning community. This project uses data mining techniques to extract data and features required for machine learning tools. Tools that are used are Anomaly Detection and Neural Networks. Anomaly Detection is used to classify data as either malicious or normal. Malicious data and normal data are passed on to Neural Networks to remove all the false predictions. This is a research project to use Machine Learning tools to provide a robust system that is flexible, adaptive, simple and deployable in real-time systems.

Key words: Network Intrusion, Neural Networks

I. INTRODUCTION

Network and system security is of paramount importance in the present data communication environment as this is an era of "Internet of Things"- An idea to connect every wired and wireless device. Threats against public as well as private networks are scaling up daily, thereby, increasing the need for Intrusion Detection Systems (IDS) on network systems throughout the corporate world. Network Intrusion Detection is a process of constantly monitoring the network to identify, block and report anomalous behavior. Function of NIDS is to safeguard distributed computing environments that are managed and controlled by a particular network. These systems focus on detecting the intrusion before the attack reaches the vicinity of the target network or computer. Network traffic is often seen to show a sudden deviation from normal behavior. Some of these aberrations are caused by malicious network attacks such as Denial-Of-Service or viruses, whereas others are the result of equipment failures and accidental outages [2]. NIDS performs thorough checks on the content of every packet, traveling through a given network in efforts to detect intrusions. This monitoring process provides better security than a mere firewall can. NIDS handles traffic and information, logging every application as it travels through a particular network and have proven to be a viable measure for securing the information of an organization. Even with prevalent intrusion detection tools and systems, such as Firewalls, unassailable security cannot be expected. Firewall is a network security system that monitors and controls the incoming and outgoing network traffic based on predetermined security rules. Current Firewalls use Signature based detection of attacks, which means that they operate in much the same way as a virus scanner, by searching for a known identity - or signature - for each specific intrusion event. Anomaly based system detects any traffic that is new or unusual.

This project uses data mining techniques to extract data and features required for machine learning tools. Tools that are used are Anomaly Detection and Neural Networks. Anomaly Detection is used to classify data as either malicious or normal. Malicious data and normal data are passed on to Neural Networks to remove all the false predictions. As a result, this system is robust. This is a research project to use Machine Learning tools to provide a robust system that is flexible, adaptive, simple and deployable in real-time systems.

Even with prevalent intrusion detection tools and systems, such as Firewalls, unassailable security cannot be expected. Firewall is a network security system that monitors and controls the incoming and outgoing network traffic based on predetermined security rules.[1]

Current Firewalls use Signature based detection of attacks, which means that they operate in much the same way as a virus scanner, by searching for a known identity - or signature - for each specific intrusion event. While signature-based IDS is very efficient at sniffing out known attacks as in anti-virus software, depends on receiving regular signature updates. As a result, signature-based IDS is good as long as its database of stored signatures is constantly updated [3].

Anomaly based system detects any traffic that is new or unusual. In network traffic terms, it captures all the headers of the IP packets running towards the network. From this, it filters out all known and legal traffic, including web traffic to the organization's web server, mail traffic to and from its mail server, outgoing web traffic from company employees and DNS traffic to and from its DNS server.

This system, using artificial intelligence tools is proposed to provide complete security against potential malicious activities when used with Firewall. This acts as an additional wall against an already secured network. The intention of this project is to overcome the issues in current intrusion detection system and provide a security system with better accuracy.

II. RELATED WORK

A deliberate attempt to enter a network and break the security of the network and thus breaking the confidentiality of the information present in the systems of the network. The person who tries to attempt such an action is called as an Intruder and the action can be termed as Network Intrusion.

A. Snort

Snort is an open source Network Intrusion detection and prevention system. It is a packet sniffer that monitors the network in real-time and detects suspicious anomalies. Through protocol analysis and content searching and matching, Snort detects attack methods, including denial of service, buffer-overflow, CGI attacks, stealth ports scans,

and SMB probes. After the attack is detected snort sends a real-time alert to a pop-up window.[4]

B. Suricata

The Suricata Engine is an Open Source Next Generation Intrusion Detection and Prevention Engine. Suricata is a rule-based ID/PS engine that utilizes externally developed rule sets to monitor network traffic and provide alerts to the system administrator when suspicious events occur. Designed to be compatible with existing network security components, Suricata features unified output functionality and pluggable library options to accept calls from other applications.[5]

C. Comparison Between Existing Systems and Proposed System

Existing systems suggest that due to false prediction of attacks, most of the algorithms based on Machine Learning have failed. Also, due to the complexity of analysis and design, these systems are not deployed. Another reason for not deploying such systems in real is, low accuracy due to false learning, Existing systems have not used Neural Networks and Anomaly Detection as a hybrid model.
Proposed system along with the collaboration with Firewall is expected to give better accuracy and also avoid false predictions. Compared to systems that use Signature based detection of attacks, the proposed system is adaptive and can detect variations in the known attacks that are not detected by such systems. This thus gives a better security to the user.

D. Types of Attacks Considered

There are six types of attacks which are being considered:
(i) **Smurf** is one of the kinds of Denial-of-Service attacks in which the network is flooded with spoofed ping messages which renders the system unresponsive.[6] (ii) **Neptune** attack where attacker sends session establishment packet with forged source IP address and victim machine then uses its resources and waits for session conformation which renders the victim machine unavailable to legitimate traffic. [7] (iii) **Teardrop** attack is a denial-of-service (DoS) attack that involves sending fragmented packets to a target machine. Since the machine receiving such packets cannot reassemble them due to a bug in TCP/IP fragmentation reassembly, the packets overlap one another, crashing the target network device.[8] (iv) **Buffer Overflow** is the anomaly where a program while writing the data overruns the boundaries of the buffer, which may cause the system to crash or a breach of system security. (v) **IP Sweep** attack is the one when one source IP address sends 10 ICMP [15] echo requests (or pings) to different hosts within a defined interval (5,000 microseconds is the default), and finally rejects the 10th and all further ICMP echo requests from that host for the remainder of that threshold period. (vi) **Warez Client** attack is the one which could pose a moderate security threat. It does not require immediate action.[9]

III. DESIGN AND IMPLEMENTATION

A. High Level Design

The Network Intrusion Detection System is coupled with a Firewall[14]. Firewall here acts as a wall against the network threats from the Internet while the NIDS acts as protection against attacks on the Local Area Network. Most of the malicious behavior is detected by the Firewall and only those packets that are considered safe by the Firewall is fed to the NIDS for further verification. Since Firewalls cannot detect attacks that are new to it, it lets many malicious packets to pass. These packets are then observed by NIDS to identify malicious packets using Anomaly based approach. Data obtained by Network Sensors are first extracted. This process is known as data extraction. This huge data needs to be reduced to avoid redundant and inconsistent data. For this purpose, data preprocessing is employed along with feature extraction and feature scaling. Features are then normalized. Data is now ready to be passed on to the Network Intrusion Detection System. Details of this process are discussed in the next section. The NIDS consists of two phases: Anomaly Detection Phase and Neural Networks Phase. Datasets after normalization and scaling, are passed to Anomaly Detection Phase where the datasets are classified as normal and malicious. Each of these categories of datasets are then passed to Neural Networks Phase separately. The NIDS then classifies the packets as normal or malicious. If they are claimed to be malicious, user is prompted to choose an action - Drop the packets or continue. If they are found to be normal, they are allowed to pass.

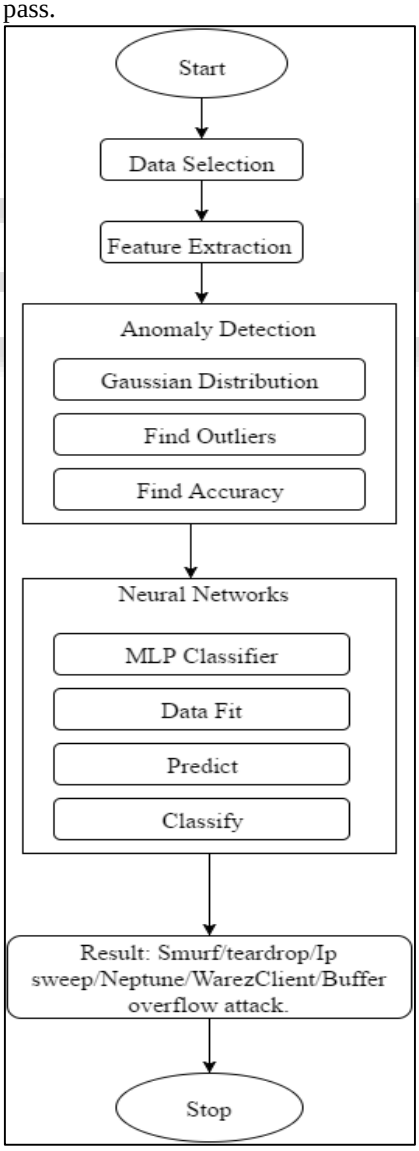

Fig. 1: Flow Diagram of proposed NIDS

B. Design Components

1) Data Extraction

a) Dimensionality Reduction:

Bandwidth and network speed are interdependent. One of them can be removed. This is called dimensionality reduction. Redundant data are also removed.

b) Data Scaling:

Scaling the attribute values belonging to each attribute to lie between a given minimum and maximum value is Data Scaling and is important as datasets might behave badly if the individual attributes do not more or less look like standard normally distributed data.

2) Data Preprocessing

Dataset:

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. The 1999 KDD intrusion detection dataset uses a version of this dataset.[10] Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local-area network

A connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows to and from a source IP address to a target IP address under some well-defined protocol. Each connection is labelled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes.

Attacks fall into four main categories:

- DOS: denial-of-service, e.g. syn flood;
- R2L: unauthorized access from a remote machine, e.g. guessing password;
- U2R: unauthorized access to local super user (root) privileges, e.g., various ``buffer overflow'' attacks;
- Probing: surveillance and other probing, e.g., port scanning.

C. Data Selection Unit:

The number of normal data to the number of attack data is disproportionate. Therefore, we run a module to proportionately extract the dataset such that we have 6 attacks, 200 each along with 1000 normal data. This not only narrows down the domain for the system, but, also ensures that the algorithm is trained to detect all these attacks efficiently. This module is written in python.

D. Feature Extraction:

Dataset considered for training the algorithm is overwhelmed with a large number of features. The information gain of an attribute suggests how much information with respect to the classification target the attribute gives. That is, how much information was available before knowing the attribute value, and how much was available after. A common measure for the information is Shannon entropy.

Shannon Entropy is the sum of the probability of each label times the log probability of that same label which is calculated as follows:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i)$$

This value is used to calculate Information Gain as follows:

Let T denote a set of training examples, each of the form $(\mathbf{x}, y) = (x_1, x_2, x_3, ..., x_k, y)$ where $x_a \in vals(a)$ is the value of the ath attribute of example x and Y is the corresponding class label. The information gain for an attribute a is defined in terms of entropy $H()$ as follows:

$$IG(T,a) = H(T) - \sum_{v \in vals(a)} \frac{|\{\mathbf{x} \in T | x_a = v\}|}{|T|} \cdot H(\{\mathbf{x} \in T | x_a = v\}) \quad [11]$$

The mutual information is equal to the total entropy for an attribute if for each of the attribute values a unique classification can be made for the result attribute. In this case, the relative entropies subtracted from the total entropy are 0.The important features extracted from KDD cup data set are as follows along with 6 other features:

- Duration :length (number of seconds) of the connection
- Protocol type :type of the protocol, e.g. tcp, udp, etc
- Service: network service on the destination, e.g. http, telnet, etc.
- Flag: normal or error status of the connection.
- Src_bytes:number of data bytes from source to destination
- Dst_bytes:number of data bytes from destination to source

1) Multivariate Gaussian Function Module:

This module computes the probability density function of the multivariate Gaussian distribution.

Gaussian distribution: Gaussian functions are bell-shaped curves that are continuous. The node output (high/low) is interpreted in terms of class membership (1/0), depending on how close the net input is to a chosen value of average.

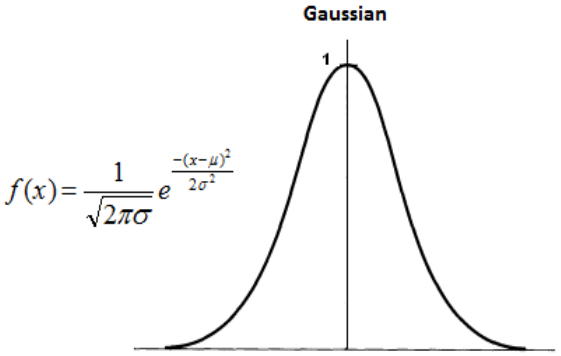

Fig. 2: Gaussian distribution Curve

2) MLP Classifier

Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function $f(\cdot) : R^m \rightarrow R^0$ by training on a dataset, where m is the number of dimensions for input and Y is the number of dimensions for output. Given a set of features $X = x_1, x_2, ..., x_m$ and a target Y, it can learn a non-linear function approximator for either classification or regression.[12]

3) The Network Intrusion Detection System

Preprocessing Unit reduces the dimensions of the dataset from (enter dimensions) to 71X15 based on Information Gain. Also, this unit reduces the number of dataset, in other words samples the dataset from around 20000 to 6093 datasets such that the training dataset has proportionate number of each attack.

Anomaly Detection Phase: Anomaly detection (or outlier detection) is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset. In NIDS this algorithm classifies normal behavior from malicious behavior. Data with normal behavior are formed as close clusters while those with malicious behavior are present as outliers. Anomaly Detection Algorithm classifies the dataset into two categories, namely, "Normal" and "Attack". "Normal" is denoted by a 0 while "Attack" is denoted by a 1.

Neural Networks Phase: A feed-forward network is a non-recurrent network which contains inputs, outputs, and hidden layers; the signals can only travel in one direction. Input data is passed onto a layer of processing elements where it performs calculations. Each processing element makes its computation based upon a weighted sum of its inputs. The new calculated values then become the new input values that feed the next layer. This process continues until it has gone through all the layers and determines the output. A threshold transfer function is sometimes used to quantify the output of a neuron in the output layer. Feed-forward networks include Perceptron (linear and non-linear) and Radial Basis Function networks. Feed-forward networks are often used in data mining.

This NIDS is proposed to have four layers - an Input Layer, two Hidden Layers and an Output Layer. During the training phase, algorithm is given the training datasets since training is supervised. After completion of training, the algorithm is fed with test dataset to find the errors. Then, Backpropagation is used to correct the error by appropriately altering the parameters and the cost function. Now, the model is ready for real-time data.

Corresponding dataset are passed onto Neural Networks Algorithm which classifies the "Attack" set into 6 attacks taken into consideration. The attacks taken into consideration are teardrop, smurf,ip sweep, buffer overflow, Warez client and Neptune. The user is then prompted about what the type of attack encountered.

IV. Result

Accuracy was checked using F1 score[13] for each face separately, which is calculated as follows:

F1 = 2 * (precision * recall) / (precision + recall)

Where precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while recall (also known as sensitivity) is the fraction of relevant instances that are retrieved. Precision and recall are defined mathematically as:

Precision = True Positives / (True Positives + False Positives).

Recall = True Positives / (True Positives + False Negatives).

F1 score for Anomaly Detection phase was found to be 95.2% and that of Neural Networks Phase was found to be 97%.

Accuracy found using weighted average of the two F1 scores based on the true results was found to be 96.4%.

V. Conclusion

The algorithm shows good results on datasets with equally or proportionately numbered attacks. It can be inferred that if training is done on an equally distributed dataset, the algorithm shows good accuracy, thus, making it deployable. Also as it learns through the time, it is effective in detecting variations of known attacks. When used with firewall, this can give a foolproof solution to the Network Intrusion issue.

References

[1] Boudriga, Noureddine (2010). Security of mobile communications. Boca Raton: CRC Press. pp. 32–33. ISBN 0849379423.
[2] Chundong Wang, Quancai Deng, Qing Chang,Hua Zhang and Huaibin Wang " A New Intrusion Detection System Based on Protocol Acknowledgement" IEEE 2010
[3] http://www.scmagazine.com/signature-based-or-anomaly-based-intrusion-detection-the-practice-and-pitfalls/article/30471/
[4] Rafeeq Ur Rehman, "Intrusion Detection Systems with Snort Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID", © 2003 Pearson Education, Inc. ISBN 0-13-140733-3
[5] Joshua S. Whitea, Thomas T. Fitzsimmonsb, Jeanna N. Matthewsc, "Quantitative Analysis of Intrusion Detection Systems: Snort and Suricata" Clarkson University, Potsdam, NY USA, 2013
[6] Craig A. Hugen, The latest in denial of service attacks: "Smurfing". Description and information to minimize effects. [Online document] Available: http://www.pentics.net/denial-of-service/whitepapers/smurf.cgi
[7] Dima Novikov, Roman V. Yampolskiy, Leon Reznik, "Traffic Analysis Based Identification Of Attacks", International Journal of Computer Science and Applications, ÓTechnomathematics Research Foundation Vol. 5, No. 2, pp 69 – 88
[8] Sílvia Farraposo, Laurent Gallon, Philippe Owezarski, "Network Security and DoS Attacks", 2005
[9] Dafydd Stuttard Marcus Pinto, "The Web Application Hacker's Handbook - Finding and Exploiting Security Flaws", ISBN: 978-1-118-02647-2
[10] (1998). DARPA, Intrusion Detection Evaluation. MIT Lincoln Laboratory, Available at: http://www.ll.mit.edu/ist/ideval
[11] [11] Lior Rokach, Oded Maimon, "Decision Trees". [Online Document] Available: https://datajobs.com/data-science-repo/Decision-Trees-[Rokach-and-Maimon].pdf
[12] http://scikit-learn.org/dev/modules/neural_networks_supervised.html, © 2010 - 2016, scikit-learn developers (BSD License)
[13] https://en.wikipedia.org/wiki/Precision_and_recall, May 2016
[14] Wilson, J. "The Future of the Firewall."Business Communications Review, May 2005.
[15] The ICMP Header. [Online document] Available: http://blog.csdn.net/xuhx/archive/2008/04/16/2297266.aspx

