Dos attack identifying in intrusion detection system using WinPcap

Mr.Naveen Kumar.T
dept.Computer Science
Mangayarkarasi college of Engineering
Madurai, India

Mr.Surya.G
dept.Computer Science
Mangayarkarasi college of Engineering
Madurai, India

Mr.Muthu Irulan.P
dept.Computer Science
Mangayarkarasi college of Engineering
Madurai, India

Mr.Karthick.P
dept.Computer Science
Mangayarkarasi college of Engineering
Madurai, India

Mrs.Rama Priya.M
dept.Computer Science(Assistant Professor)
Mangayarkarasi college of Engineering
Madurai, India

***Abstract*—Improving the performance of Intrusion detection system (IDS), here WinPcap package is used. The main aim of the paper is to detect the DOS attack both in online and offline. In online using windows packet capture. The attacks occur on website are captured to detect unwanted attempts at accessing, manipulating , mainly through a network, such as the Internet. These attempts may take the form of attacks such as crackers, malware or disgruntled employees. An interruption recognition framework is utilized to identify a few kinds of malignant practices that can bargain the security and trust of a PC framework. This incorporates organize assaults against defenseless administrations, information driven assaults on applications, have based assaults, for example, benefit acceleration, unapproved logins and access to delicate records, and malware (infections, Trojan ponies, and worms).**

***Index Terms*—Security, DOS Attack, Intrusion Detection.**

I. Introduction

A network intrusion detection system (NIDS) is an intrusion detection system that endeavors to recognize malicious activity such as denial of service attacks, port scans or even attempts to break into computers by monitoring network traffic. The NIDS does this by perusing all the approaching parcels and attempting to discover suspicious examples. If, for example, countless association solicitations to an extensive number of various ports are watched, one could assume that there is someone committing a ”port scan” at some of the computers in the network. It also (mostly) attempts to distinguish approaching shellcodes in a similar way that a conventional interruption recognition frameworks does. A NIDS is not constrained to investigating incoming network traffic only.Often valuable information about an ongoing intrusion can be learned from outgoing or local traffic as well. Some attacks might even be staged from the inside of the monitored network or network segment, and are therefore not regarded as incoming traffic at all. Often, arrange interruption location frameworks work with different frameworks also. They can for instance refresh a few firewalls' boycott with the IP locations of PCs utilized by (suspected) wafers.

II. Related Work

Since Denning proposed the first intrusion detection model [7], scholars have applied a variety of methods for intrusion detection. In recent years, methods ranging from relatively simple statistical methods to advanced machine learning and data mining methods have been used in attempts to extract specific patterns from network intrusions. Thereby, we can distinguish attack traffic from normal traffic, and finally build IDSs. The use of a single machine learning algorithm has inherent limitations. In recent years, different learning algorithms have been combined for a better performance.

Ibrahim et al. [8] used an unsupervised ANN to construct an IDS based on anomaly detection. The system employed self-organization map (SOM) ANNs for detection and to distinguish attack traffic from normal traffic. The detection rate can reach 92.37 percent on the KDD 99 dataset and 75.49 percent on the NSL-KDD dataset. SOMs are more powerful than static networks because dynamic networks have memory, which can be trained to learn sequential or time-varying patterns.

A classifier called GPSVM based on SVM and genetic programming (GP) was proposed by Pozi et al. [9] to improve the rare attack detection rate. Experimental results showed GPSVM can produce a more balanced classification accuracy on the NSL-KDD dataset without the need for resampling or feature selection techniques.

A hybrid method combining SVM and genetic algorithm (GA) was proposed by Aslahi-Shahri et al. [10]. The hybrid algorithm was used to reduce the number of features from 45 to 10, and the GA algorithm assigned these features into three priorities. As a result, it showed an outstanding true positive value and low false positive value on the KDD 99 dataset.

Hussain et al. [11] proposed a two-stage hybrid classification method. In the first stage, SVM was used for anomaly detection, while in the second stage, ANN was used for misuse detection. The main idea was to combine the advantages of each method to improve classification accuracy. Simulation results based on the NSL-KDD dataset demonstrated that this method outperforms individual classification of SVM and ANN algorithms.

Bamakan et al. [12] proposed a time varying chaos particle swarm optimization (TVCPSO) to set parameters and select features simultaneously for multiple criteria linear programming (MCLP) and SVM. Empirical results showed that this method achieves a high detection rate and a low false alarm rate.

Feng et al. [13] proposed an SVM method with clustering based on self-organized ant colony network (CSOACN) to combine the advantages of both methods and obtained a better classification rate and run-time efficiency. Evaluation on the KDD 99 dataset showed this algorithm outperforms SVM alone or CSOACN alone in terms of both classification accuracy and run-time efficiency.

Aburomman and Reaz [14] proposed an ensemble construction method which combines opinions from several experts into one model. Weighted majority voting (WMV) was used to improve accuracy and the best results were obtained with particle swarm optimization (PSO). However, such classifiers are based on binary classification methods which can distinguish between only two states.

An ensemble method based on bat algorithm (BA) was proposed in [15] which applied an ELM as the base classifier and the BA to optimize the original ensemble, then applying it to intrusion detection. Although the performance of the ELM is unstable, the method combining different ELMs into ensembles achieved better performance than using a single ELM. From the above work we conclude that the idea of combining different learning algorithms has achieved better results in improving the performance of the classifier. However, the classic classification algorithms need to preprocess the data and extract the features manually, and cannot adjust the parameters autonomously. To complete the goal of learning and classification, experts are needed to be involved from time to time. As a new hotspot in the study of neural networks, deep learning has attracted much attention in academia and industry as it has significant advantages in many areas.

In the intrusion detection area, deep learning has achieved some good results. Ma et al. [16] adopted spectral clustering (SC) to extract the features from the network traffic and used a multilayer deep neural network (DNN) to detect attack types. Experimental results showed that SC-DNN performs better than SVM, back propagation neural network (BPNN), random forest (RF) and Bayesian methods, with the best accuracy rates. However, the weight parameters and thresholds of each DNN layer need to be determined empirically rather than through rigorous mathematical theory.

Kang and Kang [17] proposed an efficient IDS based on DNN for in-vehicle networks. The system uses DNN to provide the probability of each class discriminating normal and attack packets in a controller area network (CAN) bus. To take advantage of deep learning, the system initializes the parameters through pre-training deep belief networks (DBN), resulting in an improvement in detection accuracy.

Erfani et al. [18] proposed a hybrid model that coupled a deep belief network (DBN) with a one-class SVM. An unsupervised DBN was trained to extract generic underlying features, while a oneclass SVM was trained from the features learned by the DBN. This model provided an efficient, accurate and scalable anomaly detection approach that is suitable for large-scale and high-dimensional domains.

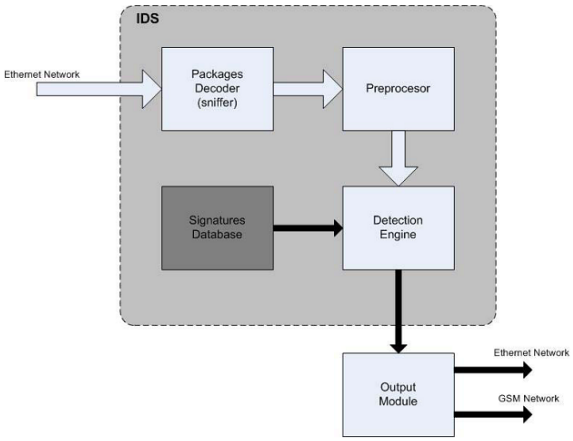
A deep learning technique called self-taught learning (STL) was used by Javaid et al. [19] to build a network intrusion detection system (NIDS). Sparse auto encoder and softmax regression based NIDS were implemented. Experiments on the NSL-KDD dataset showed that the performance of STL was comparable to the best results achieved in several previous studies.

Staudemeyer [20] claimed that, for the first time, a long short term memory (LSTM) recurrent neural network had been applied to intrusion detection, and he confirmed its effectiveness. LSTM can learn to look back in time and discover associations from a time perspective. The proposed classifier was effective in the detection of denial of service (DOS) attacks and network probes, both of which had a distinctive time series of events. Experiments showed that LSTM outperformed the winning entries of the KDD Cup 99 challenge, with 93.82 percent accuracy.

III. Proposed Solution

The IDS which we will do is the Signature Based IDS. Here we will design a system which detects the signatures. Usually the signatures are installed in the Packet and are sent to the customer framework to decimate the machines. Now we have to find out these signatures using the snort rules. Information about these signatures is used to create Snort rules. Snort's detection system is based on rules. These rules in turn are based on intruder signatures. Snort rules can be utilized to check different pieces of a data packet For the comparison of content we are using Boyer-Moore Algorithm. we will capture the packets using winpcap and Jpcap softwares. We will use James Server to activate the SMTP and POP3 protocols. We create a report for every one of the conventions which are running. We will generate a log files.

IV. Architecture Diagram



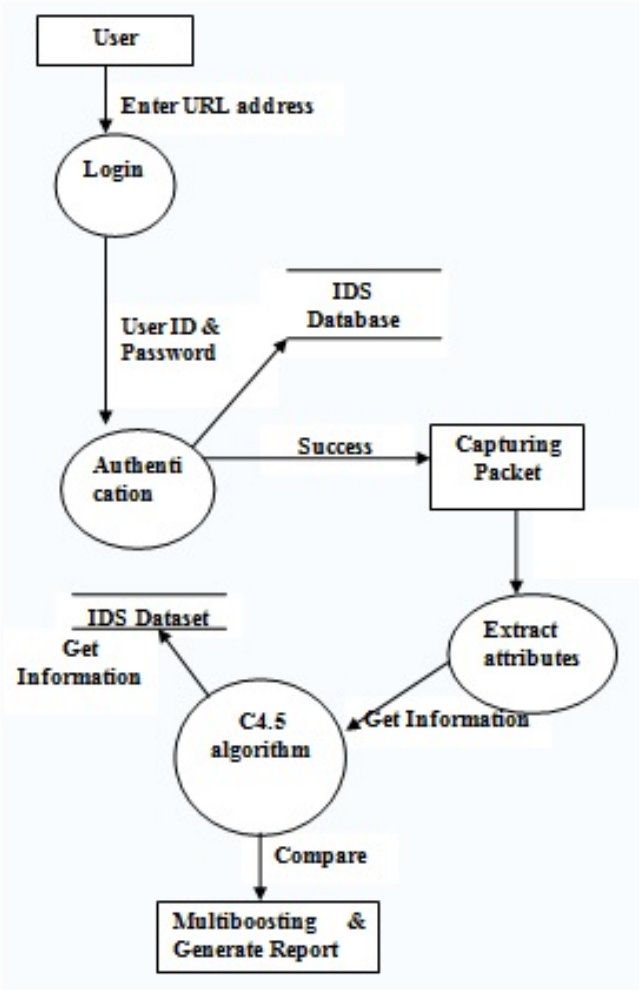
The detection engine processes rule headers and rule options differently. The detection engine builds a decision tree. The nodes of the tree test each incoming packet for increasingly precise signature elements. A packet is tested to see whether it is TCP; if so, it is passed to the portion of the tree that has rules for TCP. The packet is then tested to see whether it matches a source address in a rule; if so, it passes down the corresponding rule chains. This process happens until the packet either matches an attack signature or tests clean and is dropped. The important thing to remember is that Snort commences testing a packet after it has found a signature to match to the packet. Even if the packet could possibly match another signature, the detection engine moves on to the next packet. This is why it is valuable to organize rules so that the most malicious signatures are loaded first.

V. Experiments and analysis

A. Analyze the dataset

A data set is a collection of data, it will be presented in tabular form. Each column represents a particular variable.Each line compares to a given individual from the informational collection being referred to. It records values for each of the variables, for example, height and weight of an object or values of random numbers.Each value is known as a datum. The data set may involve data for one or more members, corresponding to the number of rows. The values may be numbers, such as real numbers or integers, for example representing a person's height in centimeters, but may also be nominal data (i.e., not consisting of numerical values), for example representing a person's ethnicity.More generally, values may be of any kinds of described as a level of measurement. For each variable, the values will normally all be of the similar kind. However, there may also be "missing values", which need to be indicated in some way.

VI. Block Diagram



A. preprocessing

In this module we will get the network packet and extract attributes using the WinPcap and JPCap. In information technology, a packet is a designed unit of data carried by a packet mode computer network. Computer communications links that do not support packets, for example traditional point-to-point telecommunications links, just transmit data as a series of bytes, characters, or bits alone. When data is designed into packets, the bitrate of the communication medium can better be shared among users than if the network were circuit exchanged. By utilizing packet switched networking it is also harder to guarantee a lowest possible bitrate.

A packet consists of two kinds of data: control information and user data (otherwise called payload). The control information provides data the network needs to convey the user data, for example: source and destination addresses, mistake recognition codes like checksums, and sequencing information. Commonly, control information is found in packet headers and trailers, with user data in between. Different communications protocols use different conventions for recognizing the components and for designing the information.

In Binary Synchronous Transmission, the packet is arranged in 8-bit bytes, and special characters are used to delimit the different components. Other protocols, similar to Ethernet, set up the start of the header and data elements by their area with respect to the start of the packet. Some protocols format the information at a bit level rather than a byte level.

A good analogy is to consider a packet to be like a letter: the header is like the envelope, and the data area is whatever the person puts inside the envelope. A difference, however, is that some networks can break a larger packet into smaller packets when fundamental (note that these smaller data elements are still organized as packets). A network design can achieve two major results by using packets: mistake identification and multiple host addressing. A nominal traffic profile consists of single and joint distributions of different packet attributes that are considered unique for a site. Candidate packet attributes from

IP headers are: 1. packet size, 2. Time-to-Live (TTL) values, 3. protocol-type values, and 4. source IP prefixes. 5. TCP flag patterns and 6. server port numbers, i.e., the smaller of the sourceport number and the destination port number.

Server port number is more stable than sort/goal port numbers because most of the well-known port numbers are small numbers (e.g., beneath 1,024) and a large portion of Internet traffic utilizes the well-known port numbers. To build the quantity of characteristics, we can employ joint distributions of the portion of packets having various combinations, such as:

7. ¡packet-size and protocol-type¿, 8. ¡server port number and protocol-type¿, and 9. ¡source IP prefix, TCP flags and packet size¿, etc.

Joint flows consistently better address the uniqueness of the traffic distribution for a site, and are all the more determinedly to guess for the attackers. A similar number of different combinations of single attributes as required may be used while the storage space permits.

B. Data Mining Using Binary Classifier (C4 Algorithm)

Binary classifiers are generated for each class of occasion utilizing significant highlights for the class and classification algorithm .Binary classifiers are derived from the preparation test by considering all classes other than the current class as other, e.g., Cnormal will consider two classes: normal and other. The purpose of this phase is to select different features for different classes by applying the information gain or gain ratio in order to identify relevant features for each binary classifier.

Moreover, applying the information gain ratio will return all the features that contain more information for separating the current class from all other classes. The result of this ensemble of binary classifiers will be decided using arbitration function based on the confidence level of the output of individual binary classifiers

C. Multi Boosting

The impact of consolidating distinctive classifiers can be explained with the theory of bias-variance decomposition. Bias refers to an error due to a learning algorithm while variance refers to an error due to the scholarly model. The all out expected mistake of a classifier is the entirety of the inclination and the change. so as to reduce bias and variation, some troupe approaches have been presented: Adaptive Boosting (AdaBoost) Bootstrap Aggregating (Bagging),Wagging and Multiboosting. This is reason the thought idea emerged of combining both in order to profit from the advantages of both algorithms and obtain an overall error reduction.

VII. Conclusion

In this paper, we propose a new data-mining based approach by combining multiboosting and an ensemble of In the first developing world computers are really a great boon to humanity . computers solve many complicated problems easily. The project entitled A NEW DATA MINING BASED APPROACH FOR NETWORK INTRUSION DETECTION is very much use full to the user to optimizing the facing problems surrounding This approach consists of three major functions: generation of accurate binary classifiers by applying different features for different types of attacks,and a new ensemble approach of the binary classifiers for removing bias and applying multi boosting for reducing both bias and variance.The software serves as tool in facilitating tedious task of manager easier and compact. The software is to reduce the strain , which the concern having travels has to take.

References

[1] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, Intrusion detection system: A comprehensive review, Journal of Network and Computer Applications, vol. 36, no. 1, pp. 1624, 2013.

[2] S. Sharma and R. Gupta, Intrusion detection system: A review, International Journal of Security and Its Applications, vol. 9, no. 5, pp. 6976, 2015.

[3] J. Allen, A. Christie, W. Fithen, J. McHugh, and J. Pickel, State of the practice of intrusion detection technologies, DTIC Document, Tech. Rep., 2000.

[4] G. E. Hinton, S. Osindero, and Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural computation, vol. 18, no. 7, pp. 15271554, 2006.

[5] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, DRAW: A recurrent neural network for image generation, ICML, 2015.

[6] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates et al., Deep speech: Scaling up end-to-end speech recognition, arXiv preprint arXiv:1412.5567, 2014.

[7] D. E. Denning, An intrusion-detection model, IEEE Transactions on software engineering, no. 2, pp. 222232, 1987.

[8] L. M. Ibrahim, D. T. Basheer, and M. S. Mahmod, A comparison study for intrusion database (KDD99, NSL-KDD) based on self organization map (SOM) artificial neural network, Journal of Engineering Science and Technology, vol. 8, no. 1, pp. 107119, 2013.

[9] M. S. M. Pozi, M. N. Sulaiman, N. Mustapha, and T. Perumal, Improving anomalous rare attack detection rate for intrusion detection system using support vector machine and genetic programming, Neural Processing Letters, vol. 44, no. 2, pp. 279290, 2016.

[10] B. Aslahi-Shahri, R. Rahmani, M. Chizari, A. Maralani, M. Eslami, M. Golkar, and A. Ebrahimi, A hybrid method consisting of GA and SVM for intrusion detection system, Neural Computing and Applications, vol. 27, no. 6, pp. 16691676, 2016.

[11] J. Hussain, S. Lalmuanawma, and L. Chhakchhuak, A two-stage hybrid classification technique for network intrusion detection system, Interna tional Journal of Computational Intelligence Systems, vol. 9, no. 5, pp. 863875, 2016.

[12] S. M. H. Bamakan, H. Wang, T. Yingjie, and Y. Shi, An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization, Neurocomputing, vol. 199, pp. 90 102, 2016.
[13] W. Feng, Q. Zhang, G. Hu, and J. X. Huang, Mining network data for intrusion detection through combining SVMs with ant colony networks, Future Generation Computer Systems, vol. 37, pp. 127140, 2014.
[14] A. A. Aburomman and M. B. I. Reaz, A novel SVM-kNN-PSO ensemble method for intrusion detection system, Applied Soft Computing, vol. 38, pp. 360372, 2016.
[15] Y. Shen, K. Zheng, C. Wu, M. Zhang, X. Niu, and Y. Yang, An ensemble method based on selection using bat algorithm for intrusion detection, The Computer Journal, pp. 113, 2017.
[16] T. Ma, F. Wang, J. Cheng, Y. Yu, and X. Chen, A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks, Sensors, vol. 16, no. 10, p. 1701, 2016.
[17] M.-J. Kang and J.-W. Kang, Intrusion detection system using deep neural network for in-vehicle network security, PloS one, vol. 11, no. 6, p. e0155781, 2016.
[18] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning, Pattern Recognition, vol. 58, pp. 121134, 2016.
[19] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, A deep learning approach for network intrusion detection system, in Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), New York, NY, USA, vol. 35, 2015, p.2126.
[20] R. C. Staudemeyer, Applying long short-term memory recurrent neural networks to intrusion detection, South African Computer Journal, vol. 56, no. 1, pp. 136154, 2015.