Article can be accessed online at http://www.publishingindia.com

Intelligent Firewall Using Intrusion Detection System Based on Neural Networks

Mandar Dinesh Sawant*, Madhura Milind Phatak, Mrugank Ajay Ranavde***, Nitya R. Laxamanan******

Abstract

Intrusion detection plays an important role in today's computers and communication technology. It is therefore very important to design an Intrusion Detection System (IDS) efficient in time that has low False Positive Rate (FPR) and False Negative Rate (FNR). A combined mechanism of packet filtering and IDS based on Artificial Neural Networks (ANNs) and rule matching is implemented to prevent network attacks with improved accuracy. An integration of the IDS with the firewall would increase the security levels to a great extent.

Keywords: Intrusion Detection, Intelligent Firewall, Neural Network

1. Introduction

Since last two decades, computer networks started to grow with tremendous speed because of which network security mechanisms are major issues. Many different security mechanisms were designed, yet none of them was able to provide security to large extent with efficiency. Firewalls were made to protect the network from attacks, but they were not capable of blocking intrusions from inside the network.

The firewall, on the system, monitors the incoming packets from the network and checks for any discrepancies, if they are found the corresponding packet is blocked. In case of an intrusion the malicious code is encrypted into the packet or some field of the packet is modified. Due to this, the firewall is unable to detect the intrusion and the machine is compromised.

The intrusion detection system scans the packets and tries to identify malicious codes in them. If a malicious packet is detected the intrusion detection system can only notify an alert, but cannot take any action.

The proposed system tries to integrate the IDS with the firewall so that an alert from the IDS will prompt the firewall to take suitable action.

The IDS will be made intelligent using a neural network. This gives the capability to the IDS to learn about newer attacks and update its directory to take actions against them.

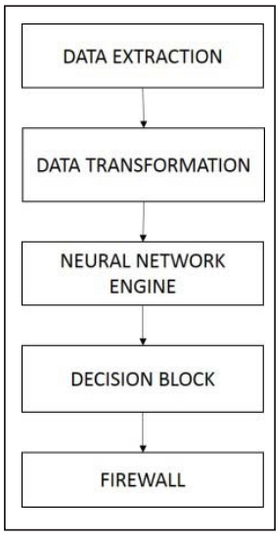
2. Architecture

The basic idea behind the proposed system will be as follows; data from the live data packets will be extracted and transformed in the form required by the neural network. This transformed data will then be given to the neural network where a specific output will be produced. This is then informed to the firewall to make suitable decisions whether to allow or block the data packet. The five blocks (as shown in Figure 1) of the system are: tha Data Extraction block, Data Transformation block, Neural

* Student, Department of Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, Maharashtra, India. Email: mandar.s@somaiya.edu
** Student, Department of Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, Maharashtra, India. Email: madhura.p@somaiya.edu
*** Student, Department of Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, Maharashtra, India. Email: mrugank.r@somaiya.edu
**** Professor, Department of Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, Maharashtra, India. Email: nityalaxmanan@somaiya.edu

Network, Action Decision block and the Firewall.

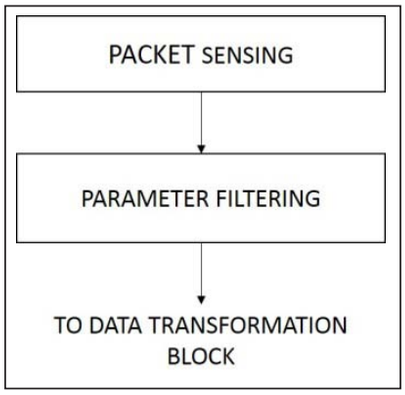
Figure 1: Basic Block Diagram



2.1. Data Extraction

This part of the system senses the incoming packets, decodes the packets and extracts all the information about the packets. The intrusion detection system does not require the values of all the fields for detecting an intrusion. It filters out the fields which are necessary for identifying an attack and gives it to the data transformation block.

Figure 2: Data Extraction Module



2.2. Data Transformation

The data transformation block transforms all the information extracted from the packet into required 41 fields mentioned in KDD Cup 1999 dataset and the relevant codes are assigned to particular fields to make it suitable for testing live packets in neural network (Selman *et al*., 2013; Bhavsar *et al*., 2013). The output is given to neural network engine.

2.3. Neural Network Engine

This part of the system consists of three major blocks (as shown in Figure 3): 1. Training block (as shown in Figure 4), 2. Actual Neural Network, and 3. Network Retraining Block (as shown in Figure 5). For training purpose we will be using reduced KDD CUP 1999 data for intrusion detection (kddcup99: online). In the training phase, the transformed data is processed so as to get it in the proper format to train the network. This is called pre-processing of data. In the pre-processing stage actions like reduction of data are performed. This pre-processed data is used to train the neural network. The weights that are given by the trained networks are saved for future use. The network takes decisions using these weights. Using these weights a set of rules is created that helps in deciding nature of the packet. Neural Network determines nature of packet and gives output whether to allow or block the packet. Purpose of the retraining block is to improve accuracy of network over the period of time and to avoid overtraining of the network (Fu, 1994).

Network retraining block consists of logging system which records live packet's transformed data and action taken by neural network on it. The re-training module periodically trains the network to include newly identified intrusions. The newly logged entries along with the previous training data set are given to the neural network for retraining.

Figure 3: Neural Network Engine

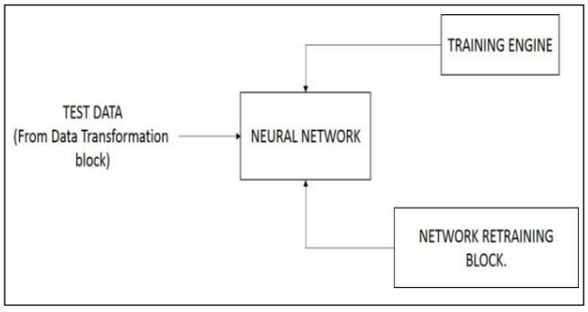
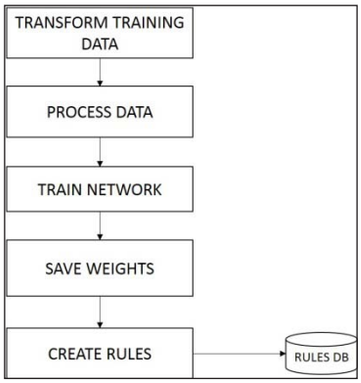


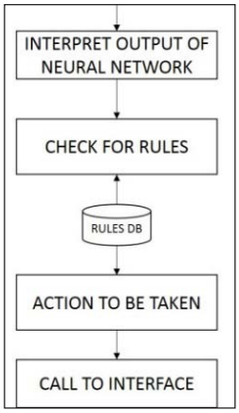
Figure 4: Training Engine



2.4. Action Decision Block

After testing the live packet, output of the neural network is given to the Action decision block (as shown in Figure 5). It interprets the output and checks for the rules in rules database created in training phase of the network. The action is given to the call interface (Huang *et al.*, 2010) which calls firewall and notifies packet filtering module to revise its rules for that particular packet.

Figure 5: Action Decision Block



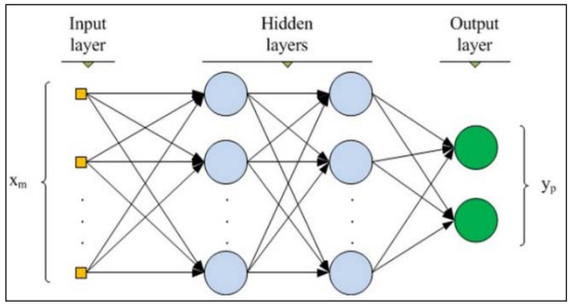
3. Types of Networks

There are many algorithms available that are useful to implement a neural network. Two of them are the Back-propagation algorithm and the Radial Basis Function algorithm. Based on these two algorithms, we have two networks one is the Multilayer Perceptron (MLP) (as shown in Figure 6) that uses back-propagation algorithm and the other is the Radial Basis Function Network (RBFN) (as shown in Figure 7) that uses Radial Basis Function algorithm.

3.1. Multilayer Perceptron

Here there are three parts: obtaining number of inputs, defining number of outputs and deciding number of hidden layers. Number of inputs is the number of features that characterise the data and number of outputs is the number of classes. Each neuron in the hidden layer is represented by an activation function. The function should be able to accept an input within any range and produce output in a strictly limited range.

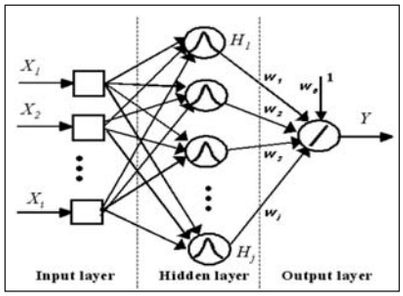
Figure 6: Multilayer Perceptron



Radial Basis Function Network

It is a two layer network with only one hidden layer whose output units form a linear combination of basis (kernel) functions computed by the hidden units. The basis functions in the hidden layer produce a localised response to the input.

Figure 7: Radial Basis Function Network



4. Comparitive Analysis

The networks were trained and tested for a 2- input XOR gate.

Input Neurons = 2

Output Neurons = 1

Stopping Criteria: Maximum error = 0.01

Learning Rate = 0.2

For MLP:

No. of neurons:

Layer1 = 5

Layer2 = 4

Layer3 = 5

Transfer Function = Sigmoid

Learning Rule = Back-propagation

No. of iterations = 2370

Total network error = 0.009995

Total MSE = 0.01836

Table 2: Output for MLP

INPUT 1	*INPUT 2*	*NETWORK OUTPUT*	*EXPECTED OUTPUT*
0	0	0.183	0
1	0	0.8613	1

For RBFN:

No. of neurons = 5
No. of iterations = 7312
Total network error = 0.009977
Total MSE = 0.0196

Table 2: Output for RBFN

INPUT 1	*INPUT 2*	*NETWORK OUTPUT*	*EXPECTED OUTPUT*
0	0	0.01555	0
1	0	0.868	1

5. Conclusion

This paper proposes a scalable model of IDS and Intelligent Firewall to provide high security. The Neural Network Engine uses MLP with Back-propagation algorithm for learning which gives high efficiency to detect an attack. The proposed system is efficient enough to detect and block the intrusions from inside the network which is not the case with firewall alone. The system not only detects and blocks predefined attacks and intrusions but also is capable of detecting and blocking new attacks by learning about them. It also saves the results for new attacks for future use.

6. Future Works

For increasing the efficiency of the proposed system, only one part of the model i.e. the Neural Network can be replaced with other efficient models like the Committee Machine, Quarantine Channels, etc. to achieve enhanced accuracy.

References

Ali, A. O., Saleh, A. I., & Badawy, T. R. (2010). *Intelligent Adaptive Intrusion Detection Systems using Neural Networks*. In Proceedings of International Journal of Video & Image Processing and Network Security.

Bhavsar, Y. B., & Waghmare, K. C. (2013). *Intrusion Detection System using Data Mining Technique: Support Vector Machine.* In Proceedings of International Journal of Emerging Technology and Advanced Engineering, March, 3(3), 581-586.

Debar, H. (2004). An Introduction to Intrusion Detection Systems.

Fu, L. M. (1994). *Neural Networks in Computer Intelligence*. New York: McGraw-Hill.

Hooper, E. (2007). *An Intelligent Intrusion Detection and Response System Using Network Quarantine Channels: Firewalls and Packet Filters*. In Proceedings of International Conference on Multimedia and Ubiquitous Engineering.

Huang, X., Wang, X., & Zhu, S. (2010). *Study on Intelligent Firewall System Combining Intrusion Detection and Egress Access Control*. In Proceedings of International Conference on Intelligent System Design and Engineering Application, (pp. 456-459).

Selman, A. H., Koker, R., & Selman, S. (2013). *Intrusion Detection using Neural Network Committee Machine*.In Proceedings of 24th International Conference on Information, Communication and Automation Technologies. (pp. 1-6).

ZongpuJia, S. L., & Wang, G. (2006). *Research and Design of NIDS Based on Linux Firewall*. In proceedings of International Symposium on Pervasive Computing and Applications. (pp. 556-560).

Web References

http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

http://neuroph.sourceforge.net/download.html