Network Intrusion Detection System Using Neural Networks

Jimmy Shum and Heidar A. Malki [1]
Senior member
University of Houston
College of Technology
malki@uh.edu

Abstract

This paper presents a neural network-based intrusion detection method for the internet-based attacks on a computer network. Intrusion detection systems (IDS) have been created to predict and thwart current and future attacks. Neural networks are used to identify and predict unusual activities in the system. In particular, feedforward neural networks with the back propagation training algorithm were employed in this study. Training and testing data were obtained from the Defense Advanced Research Projects Agency (DARPA) intrusion detection evaluation data sets. The experimental results on real-data showed promising results on detection intrusion systems using neural networks.

.

1. Introduction

While internet attacks are on the rise, the rate at which these attacks are increasing is alarming. According to the Computer Emergency Response Team (CERT), the number of reported incidents has increased to 137,529 in 2003 from 252 in 1990 [1]. CERT is located at Carnegie Mellon University's Software Engineering Institute in Pittsburg, Pennsylvania. CERT now works in conjunction with the "Department of Homeland Security's National Cyber Security Division and its US-CERT to coordinate responses to security compromises; identify trends in intruder activity; identify solutions to security problems; and disseminate information to the broad community" [1]. Table 1 shows the number of incidents per year from 1993 to 2003. It is also important to note that because attacks have become so common thanks to the widespread use of attack tools that CERT will no longer be publishing the number of incidents reported per year but rather work with the community to publish an E-crime watch survey [1].

Also, according to the IBM Global Business Security Index, "a monthly report that assesses, measures and analyzes global network security and business threats and attack trends", there was a 55% increase in network attacks from July 2004 to August 2004 [2]. The index is determined by using historical and present data by IBM's 2700 professionals and half a million monitored devices.

Table1: CERT incident report

Year	93	94	95	96
Incidents	1,334	2,340	2,412	2,573

Year	97	98	99	00
Incidents	2,134	33,734	99,859	21,756

Year	01	02	03
Incidents	52,658	82,094	137,529

Internet attacks can come in many different forms thus it is essential that there are many types of network defense systems. This project will focus on an intrusion detection system based on neural networks.

2. Internet Attacks

An Internet attack can be defined as the misuse of a system or systems without permission. The person performing the attack is called a hacker or cracker. These intruders can really be classified into two categories: 1) Outsiders – these are people from outside the network that do not have any rights to it [3]. They could be defacing a website, relaying spam, etc. and 2) Insiders – these are people from inside the network who are misusing their privileges [3]. They could be changing information, stealing company information, etc. There are several ways in which these

978-0-7695-3304-9/08 $25.00 © 2008 IEEE
DOI 10.1109/ICNC.2008.900

intruders can access a network. Physical intrusion is when a hacker physically accesses a machine. System intrusion is when the hacker is assumed to already have a low-privilege account in the system. And remote intrusion is when the hacker tried to penetrate the system remotely across the network with no access permissions at all [3].

Attacks can be classified into three types: 1) Reconnaissance – these attacks involve the gathering of information about a system in order to find its weaknesses such as port sweeps, ping sweeps, port scans, and Domain Name System (DNS) zone transfers [3], 2) Exploits – these attacks take advantage of a known bug or design flaw in the system, and 3) Denial-of-Service (DoS) – these attacks disrupt or deny access to a service or resource. It does not access any information on the system but is rather an act of vandalism.

3. Existing Intrusion Detection Systems

Intrusion Detection Systems (IDS) monitor the network for any unusual activity to determine whether it has been compromised or not. There is a distinction between a host-based and network-based ID. A host-based Intrusion Detection System runs on a single machine and monitors its own traffic for attack. A network-based Intrusion Detection System is located on an independent machine watching the activity of the entire network [3]. Intrusion can be detected using two different methods: Misuse intrusion detection and anomaly intrusion detection. Misuse, or knowledge-based, intrusion detection is the more common technique where it compares the network traffic against a database of known attacks [4]. An alarm is set off when an event matches the signature of an attack in the database. Anomaly, or behavior-based, intrusion detection analyses the network traffic for any deviation from the normal or expected behavior of the system [3]. It then learns and adapts from that information. One of the advantages of misuse intrusion detection is its accuracy and the return of low false positives. A false positive is when the intrusion detection system misinterprets normal traffic for an attack. The disadvantage of misuse intrusion detection is keeping the database up-to-date with regular maintenances. The high return of false positive is the main disadvantage of the anomaly intrusion detection system. This is the result of its ability to change and adapt over time. But one of its advantages is the ability to "detect attempts to exploit new and unforeseen vulnerabilities" even contributing to the discovery of new attacks [4].

There are different types of techniques that are applied to the data that reaches the Intrusion Detection System. The descriptions of some of the most common approaches are given below, including the neural network approach used in this study:

Expert Systems – data is compared to an audit trail with a previously predefined set of rules describing the attack.

Signature Analysis – similar to the expert system approach it coverts the "semantic description of an attack into the appropriate audit trail format" [5]. This is one of the most common methods used in commercial systems such as Stalker, Real Security, and Cisco IDS.

Colored Petri Nets – generates graphical representation of attacks using expert knowledge bases.

Statistical Analysis – the behavior of the data is compared to a number of variables over time. Some examples of these variables are: user login, usage of disk space, memory, CPU, etc.

Data Mining – excels at extracting "previously unknown but potentially useful data from large stores of data" [5].

Neural Networks – input and output vectors are compared using a learning algorithm.

In particular several neural networks-based approaches were employed for intrusion detections. In the experiments by Jake Ryan, Meng-Jang Lin, and Risto Miikkulaine of The University of Texas at Austin, their system was 96% accurate in detecting unusual activity in their network[12]. In another project by Robert Birkely of Griffin University, a classification rate of 100% for normal, 92% for known attacks, and 80% for unknown attacks was achieved [13].

Kohonen's Self-Organizing Map was also used for intrusion detections. Unlike backpropagation, self-organizing maps (SOM) are unsupervised learning networks and the actual outputs are not known. Experiments using SOMs have also been very successful in categorizing normal traffic from malicious traffic. Brandon Rhodes, James Mahaffey, and James Cannady were able to classify all normal traffic within a distance of zero to three. Whereas, attack sessions were classified at a distance of eighty to six hundred thirty - indicating an extreme anomaly from the normal pattern [14].

4. Preprocessing and Architecture

In this work, a feedforward neural network based on the backpropagation training algorithm is used. The architecture of the proposed neural network consists of an input layer, a hidden layer, and an output layer. The number of input nodes will be determined from the input data set. The number of nodes in the hidden layer will be varied throughout the experiment. There are two nodes in the output layer to distinguish between normal and attack traffic.

The data for both training and testing were obtained from the DARPA depository. The MIT Lincoln Laboratory performed experiments during 1998, 1999, and partially during 2000. Data sets for this project were taken from week 4 of 1998. During 1998, seven weeks of training data and 2 weeks of testing data were recorded. Each week's data was also broken down into days of the week. Normal network traffic as well as attack traffic is included in the data sets. The attacks are broken down into the following 4 different categories: denial of service, remote to user, user to root, and surveillance/probe. All data from the DARPA data sets were taken by the TCPDUMP program under a UNIX environment. An example of the data is illustrated in Table 2.

Table 2: Sample data

1	06/19/1998	07:53:13	00:00:01	ntp/u	123	123	172.016.112.020	192.168.001.010
2	06/19/1998	07:53:13	00:00:01	ntp/u	123	123	172.016.112.020	192.168.001.010
3	06/19/1998	07:55:34	00:00:01	domain/u	53	53	192.168.001.010	192.168.001.020

The original parameters that were included in the TCPDUMP file are as followed:

Session ID – uniquely identifies that connection session.
Start Date – date the session was started.
Start Time – time of the day the session was started.
Duration – length of the session.
Service – the protocol used by the session.
Source Port – port used by incoming service.
Destination Port – port targeted by service.
Source IP – internet address of source object.
Destination IP – internet address of destination object.

These parameters were evaluated, preprocessed, and tested many times over the course of this project to prepare them for training neural networks. The use of real world data provided a challenge in trying to converge the network. Ultimately, the following descriptors were used: date, time, duration, protocol, source port, and destination port. Addresses were excluded because traffic could come from any source and arrive at any destination. Time and duration were represented by the hour and by the second respectively. And all port numbers greater than 1023 were represented by a zero. All ports accessed above 1023 are ports randomly generated by applications and are not part of the range of commonly used ports. Using the newly formatted input data set the network was able to converge with parameters as shown in Table 3. The network converged as shown in Figure 1 with the parameters shown in Table 3.

Table 3: Best network convergence parameters

Learning Rate	0.01
Hidden Neurons	20
Performance Goal	0.00001
Number of Epochs	20000
Transfer Functions	tansig(hidden), purelin(output)

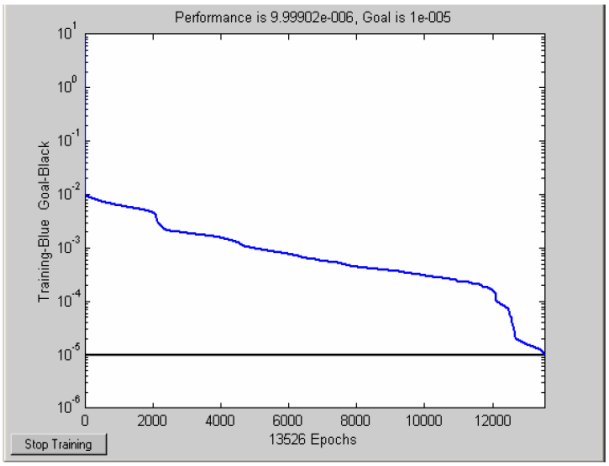

Figure 1. Network convergence

5. Testing of Training Traffic

The training data set was then tested against the trained network. The output graphs for the testing of the training data set can be seen in Figures 2 and 3. The desired output is represented in black and the actual output is represented in red. Figure 2 is first neuron of

the output and Figure 3 is the second neuron of the output. Attack traffic is represented by an output of (1, 0) whereas normal traffic is represented by an output of (0, 1). The results shows almost 100% match between the desired and actual output because the red output (desired) traced the black output (actual) through all sessions with very minimal deviation. The very small deviation of the actual output over the desired output presented a successfully trained neural network. The trained network classified all sessions from the training data set correctly.

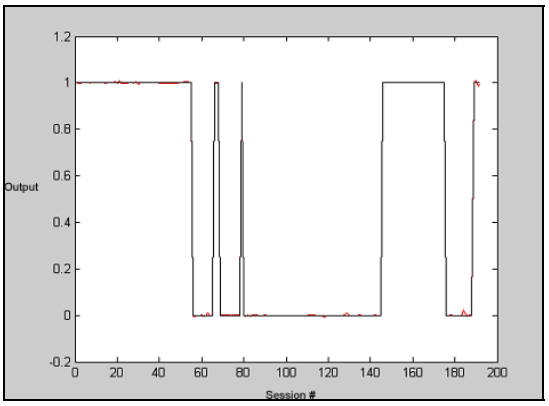

Figure 2. Neuron 1 output

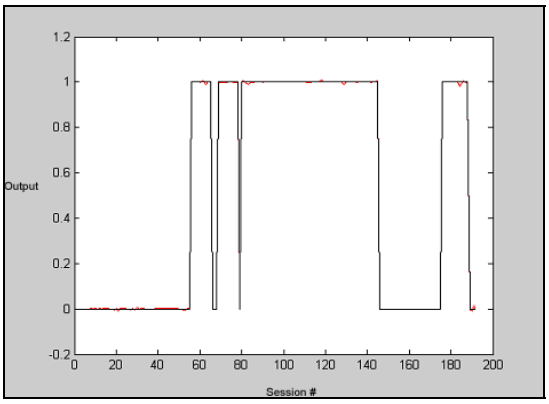

Figure 3. Neuron 2 output

6. Results and Discussions

Four separate tests were performed in order to evaluate the effectiveness of the neural network in detecting network traffic. The first experiment involved testing the same traffic that was used for training. This would give an idea of whether the network was properly trained. All of the traffic data were correctly recognized by the network as shown in Table 3. The data sets used for testing are broken down into normal traffic, unknown attack traffic, and known attack traffic. The unknown attack traffic data set is made up of data that was not used in training the network. The results indicate that the proposed neural network model classified the first three set with 100% correct classification and 76 % of unknown attacks were classified correctly. The results of all 4 experiments are shown on Table 3.

Table 3. Experimental classification rate

Experiment	Classification Results	Correct Classification
Training Set	196/196	100%
Normal Traffic Set	50/50	100%
Known Attack Set	25/25	100%
Unknown Attack Set	19/25	76%

7. Conclusions

As network attacks continue to climb to new record highs every year, new methods are being sought to prevent malicious hackers from accessing private information. In this project neural networks were used as one of the effective methods in detecting such intrusions. Their ability to learn and adapt to new data is one of the main advantages of using neural networks. The network was first trained by using data gathered by the MIT Lincoln Laboratory under the DARPA Intrusion Detection Evaluation project. Three sets of testing experiments were performed to validate the effectiveness of the proposed model in detecting network intrusions. Data sets with normal traffic, known attacks, and unknown attacks were next tested in the trained neural network. The results indicate that the proposed neural network model correctly classified all normal and known traffic set and 76 % of unknown attacks were classified correctly.

7. References

[1] http://www.cert.org , 2004.

[2] http://www.networknewz.com, 2004.

[3] http://www.infosyssec.net/infosyssec/netintrufaq.htm, 1999.

[4] http://www.sans.org/resources/idfaq/, 2004.

[5] http://www.windowsecurity.com/articles/IDS-Part2-Classification-methodstechniques.html, 2004.

[6] http://www.ll.mit.edu/IST/ideval/index.html, 2001.

[7] http://www.mathworks.com/products/matlab/, 2005.

[8] Fausett, Laurene. *Fundamental of Neural Networks.* 1st Edition, Prentice Hall, 1994.

[9]http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html,2003.

[10] http://ei.cs.vt.edu/~history/Perceptrons.Estebon.html, 1997.

[11] http://fann.sourceforge.net/report/node4.html, 2003.

[12] Jake Ryan, Meng-Jang Lin, and Risto Miikkulainen, "Intrusion Detection with Neural Networks", 1998.

[13] Robert Birkely, "A Neural Network Based Intelligent Intrusion Detection System", 2003.

[14] Brandon Craig Rhodes, James A. Mahaffey, and James D. Cannady, "Multiple Self-Organizing Maps for Intrusion Detection", 2000.