An Intelligent Approach for DOS Attacks Detection

Amr Hassan Yassin
Hany Hamdy Hussien
AN INTELLIGENT APPROACH FOR DOS ATTACKS DETECTION

AMR HASSAN YASSIN
Faculty of Engineering, University of Alexandria, Egypt

HANY HAMDY HUSSIEN
Department of Information Technology, IGSR, University of Alexandria, Egypt

ABSTRACT

The purpose of this work is to get an enhanced detection approach for the Denial of Service (DoS) attacks intrusion problem done in a particular network. An achievable optimized model of neural network is presented for the proposed detection system. The data used in training and testing is the data collected by the common packet analyzer Tcpdump. This RBF-NN model can be used as a general classifier for several types of attacking methods.

Keywords: Intrusion Detection, Neural Network, Radial Basis Function, and Denial of Service.

1. INTRODUCTION

Denials of Service (DoS) attacks are a serious threat for the prominent e-commerce internet sites such as Amazon, CNN, E*Trade, Yahoo and eBay. DoS attacks can consume memory, CPU, and network resources and damage or shut down the operation of the resource under attack (victim). The quality of service enabled networks (QoS), which offer different levels of service, is vulnerable to such DoS attacks.

Denial of service (Dos) is a type of attack in which a hacker issues a huge amount of packets to congest specific servers’ services, consequently blocking legitimate users from normal access to the services. Distributed DoS (DDoS) attacks are another form of DoS attacks in which a host or hosts suffer from receiving a huge amount of packets issued by zombies. DDoS attacks often do not rely on particular network protocols or system weaknesses [1].

Often hackers of DoS attacks spoofed their attack packets’ source addresses, and in DoS attacks, each zombie only sends a limited amount of packets to a victim or victims. Both make it very difficult to trace to the real attackers [2]. According to a 2007 CSI Computer Crime and Security Survey [3], DoS attacks were in the top 5 among all attack types. 25 percent of respondents’ computers had detected DoS attacks. These show the severity of information security.

According to a 2008 CSI Computer Crime and Security Survey [4] about attacking technologies used, intrusion detection and intrusion prevention systems are very important tools for security. This survey also described that DoS attack has severely influenced on network security in a year after year tendency of the rise.

The surveys imply a fact that although present DoS is dangerous to computer networks and systems, and many intrusion detection methods have been developed, none of current detection approaches can completely protect a system and prevent a system from DoS attacks. The key reason is hackers discover new weakness and then create new attacking methods almost every day. In this work, a proposed technique is presented to detect the Intrusion attacks, using radial basis function neural network (RBF-NN). The following parameters are introduced as the normalized training inputs for the introduced model: i) packet sequence number ii) data expected sequence number iii) next data expected sequence number and vi) number of bytes received. The following parameters are expected to be output parameters for the neural network: i) source address ii) source port iii) destination address and vi) destination port as indicated in figure 1.

2. RELATED WORK

The network-based denial-of-service intrusion detection DoSID system is introduced as input-output pairs using neural network to detect Denial of Service attacks. The training dataset is used to train and test the proposed neural network. The RBF-NN is trained to detect many forms of attack. It has been found that the neural network detects the known attacks which have been used in the training of the neural network. Also, the neural network has the ability to detect the unknown attacks which have never been used in the training phase [5].

It should be noted that, the Distributed Intrusion Detection System (DIDS) project was sponsored by the United States Air Force Crypto logic Support Center through a contract with the Lawrence Livermore National Labs. The DIDS architecture combines distributed monitoring and data reduction with centralized data analysis, but it did not scale well for large networks since addition of any new component increases the load on the DIDS director component, and the data flow from monitors to DIDS director consumes high network bandwidth [6]. Our proposed intelligent approach focuses on on-line detection of attacks and classifying the type of threat.

The Autonomous Agents for Intrusion Detection (AAFID) made use of multiple layers of agents organized in a hierarchical structure with each layer performing a set of intrusion detection tasks. Administrator can send global instructions to all agents so that network and data can be respectively monitored and analyzed on each node. Data on end nodes are collected by agents who are dispatched by local monitoring nodes. Local monitoring nodes are responsible for analyzing data gathered by agents. Global monitoring nodes are in charge of integrity monitoring. However, AAFID uses only static agents and is deprived of some of the benefits mobile agents can offer [7].
The Lightweight agents for intrusion detection had been developed for an IDS that deploys distributed multiple layers of lightweight intelligent mobile agents and applies data mining techniques to detect intrusions. An agent monitor system roams around different system networks, analyzing and integrating collected information and transferring the results to users and at last, storing the results into a database. This system allows an agent to increase its new ability during its execution period, and provides more convenient mechanisms to improve IDS’s communication capability [8].

The original method for detection of DDoS attacks has been introduced based on a statistical pre-processor and an unsupervised artificial neural net. In addition, SPUNNID system has been designed based on the proposed method. The statistical pre-processing has been used to extract some statistical features of the traffic, showing the behavior of DDoS attacks. The unsupervised neural net is used to analyze and classify them as either a DDoS attack or normal. Moreover, the method has been more investigated using attacked network traffic, which has been provided from a real environment. The experimental results show that SPUNNID detects DDoS attacks accurately and efficiently [9].

An overview of real time data mining-based intrusion detection systems (IDSs) is presented by researcher that focused on problems related to deploying a data mining-based IDS in a real time environment also discussed a distributed architecture for estimating cost-sensitive models in real time. Adaptive learning algorithms are used to improve usability that facilitates model creation and incremental update. Unsupervised anomaly detection algorithms are used to reduce the reliance on labeled data. Author (Lee et. al., 2002) gives an architecture consisting of sensors, detectors, a data warehouse, and model generation components. Presented architecture facilitates the sharing and storage of audit data and the distribution of new or updated models which improves the efficiency and scalability of the IDS [10].

3. NEURAL NETWORK ARCHITECTURE

The Radial basis functions (RBF) neural network represents a relatively new model of neural network. On the contrary to classical models (multi layer perceptions, etc.) it is a network with local units which was motivated by the prevalence of many local response units in human brain. Other motivation came from numerical mathematics, RBF were first introduced in the solution of real multivariate problems [11].

The input into an RBF network is nonlinear while the output is linear. Due to their nonlinear approximation properties, RBF networks are able to model complex mappings, which perception neural networks can only model by means of multiple intermediary layers.

In order to use a radial basis function network that specify the hidden unit activation function, the number of processing units, a criterion for modeling a given task and a training algorithm for finding the parameters of the network. Finding the RBF weights is called network training. If we have at hand a set of input-output pairs, called training set, we optimize the network parameters in order to fit the network outputs to the given inputs. The fit is evaluated by means of a cost function, usually assumed to be the mean square error. After training, the RBF network can be used with data whose underlying statistics is similar to that of the training set. On-line training algorithms adapt the network parameters to the changing data statistics.

Radial basis functions are embedded into a two-layer feed-forward neural network. Such a network is characterized by a set of inputs and a set of outputs. In between the inputs and outputs there is a layer of processing units called hidden units. Each of them implements a radial basis function.

Radial basis function neural networks (RBF-NN) approach the problem as a function approximation problem. The structure of RBF-NN is shown in figure 1.

This comprises of three layers: The input layer consisting of \( n \) sets of input vectors. The hidden layer has \( m \) radial basis neurons \( net_j \), and \( j = 1,2,\ldots,m \).

The output layer consists of linear neurons. The function \( net_j \) has the form:

\[
net_j = \|x_n - c_n\| \quad (1)
\]

\[
net_j(x_n) = \exp(-\frac{||x_n - c_n||^2}{r^2}) \quad (2)
\]

Which is the Euclidian distance between each input \( x_n \) and corresponding weighting vectors with gaussian functions representing the center of basis function \( net_j \), and \( r \) is a scalar value called spread constant [12].

The output of the network is expressed as:

\[
F_k(x_n) = \sum_{j=1}^{m} w_{kj} net_j(x_n) + b_j \quad (3)
\]

Where \( w_{kj} \) and \( b_j \) are the weighting factors of the output layers and biases respectively. Training of RBF-NN involves calculating the optimal values of the centers,
weights, and biases, which satisfies the condition of minimum sum of squared errors.

Initially the radial basis layer has no neurons. The following steps are repeated until the network's mean squared error falls below goal.

- The network is simulated.
- The input vector with the greatest error is found.
- A radial basis neuron is added with weights equal to that input vector.
- The purelin layer weights are redesigned to minimize error.

4. DATASET RESEARCH

The network Intrusion traffic was collected via tcpdump packet sniffer. It is able to capture traffic that passes through a machine. It operates on a packet level, meaning that it captures the actual packets that fly in and out of your computer. It can save the packets into a file. Tcpdump prints out the headers of network packets that pass by the network interface of the host executing tcpdump. The host used for this data collection was connected between the Enterprise LAN and external networks. Therefore, all network traffic passing between the Enterprise LAN and external networks was captured by tcpdump. Because tcpdump prints out only header information, no user data was printed [13].

When executing tcpdump, several filters can be specified. With filters specified, tcpdump will only collect data that can pass through those filters. For the purposes of these tests, filters were established so that only Internet Transmission Control Protocol (TCP) and Internet User Datagram Protocol (UDP) packets were collected. For each TCP packet, tcpdump prints the following information:

- Time Stamp
- Source IP Address
- Source Port
- Destination IP Address
- Destination Port
- Flags (Syn, Fin, Push, Rst, Or.)
- Data Sequence Number of this Packet
- Data Sequence Number of the Data Expected in Return.

For each UDP packet, tcpdump prints the following information:

- Time Stamp
- Source IP Address
- Source Port
- Destination IP Address
- Destination Port
- Length of the Packet

To protect the identity of the hosts that were communicating with each other while the network traffic was collected, all IP addresses have been modified. Each external host is assigned a "fake" IP address. All internal hosts (hosts on the Enterprise LAN) will share the same "fake" IP address.

5. SIMULATION RESULTS

Experiment was done to see performance of artificial neural network in order to model intrusion attacks of the network. The MSE (mean square error) performance is evaluated on a separate test set of 174 random samples measured at interval of 62 samples during training. Figure 1 shows the MSE learning curves of the RBF with estimated error 0.0194 for detection the intrusion attacks through many iteration epochs. The convergence curve of the proposed neural network is indicated in Figure 2. To test the performance of the RBF-NN model a linear regression procedure is performed for the original test data and the simulated output from the RBF-NN model. The regression analysis between the actual data and the simulated values for the training dataset are shown in figure 3. The best linear fit is indicated by a dashed line. The perfect fit (output equal to target) is indicated by the solid line.

The regression analysis shows a good prediction result represented by the R-value which is 0.97 as shown in Figure 3. The choice of spread constant for RBF-NN is an important because it affects the simulation results.

Table 1 indicate some chosen values in our simulation and the related simulation error.

<table>
<thead>
<tr>
<th>Spread</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spread = 0.6</td>
<td>4.1393</td>
</tr>
<tr>
<td>Spread = 1</td>
<td>1.83</td>
</tr>
<tr>
<td>Spread = 2</td>
<td>3.15</td>
</tr>
<tr>
<td>Spread = 4</td>
<td>6.13</td>
</tr>
<tr>
<td>Spread = 6</td>
<td>8.27</td>
</tr>
</tbody>
</table>

The result of neural network for the test data and their corresponding error for address and port of the source node are shown in Table 2.
Table 2. The Test Cases and Their Corresponding Error

<table>
<thead>
<tr>
<th>Source Address Actual</th>
<th>Source Address Predicted</th>
<th>Error %</th>
<th>Source Port Actual</th>
<th>Source Port Predicted</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.96</td>
<td>4</td>
<td>110</td>
<td>109.96</td>
<td>0.04</td>
</tr>
<tr>
<td>1</td>
<td>0.96</td>
<td>4</td>
<td>110</td>
<td>109.97</td>
<td>0.03</td>
</tr>
<tr>
<td>3</td>
<td>2.98</td>
<td>0.67</td>
<td>2210</td>
<td>2209.98</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>1.98</td>
<td>1</td>
<td>3209</td>
<td>3208.99</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>1.97</td>
<td>1.5</td>
<td>4051</td>
<td>4050.97</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>1.99</td>
<td>0.5</td>
<td>3164</td>
<td>3163.96</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>1.96</td>
<td>2</td>
<td>3086</td>
<td>3085.97</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Figure 3. Regression Curve for Original Data and Simulated Data

6. CONCLUSION

An introduced approach for detection the intrusion attacks is introduced which done in well defined network using an intelligent technique. The neural network was able to classify the patterns of attack occur, the neural network is training by RBF algorithm with estimated MSE 0.0194. Based on dataset collected by a “tcpdump”, a packet capture for real-time collection of data as it travels over a network. The calculations have shown that the coefficients of correlation calculated for both of the two data sets were closer to unity and the RMS (root mean square) error is less than 2% for RBF-NN model with spread=1. In this work the modelling results were realized using MATLAB release 14 from Mathworks cooperation [9], and the time of executing the training cycle until convergence took about 2 minutes on a conventional 2.4 GHz PC for RBF-NN based model.

REFERENCES:


