Hybrid Intelligent Systems for Network Security

J. Lane Thames
Georgia Institute of Technology
210 Technology Circle
Savannah, GA 31407
912-966-6874
lane.thames@gtsav.gatech.edu

Dr. Randal Abler
Georgia Institute of Technology
210 Technology Circle
Savannah, GA 31407
912-966-7933
rabler@gtsav.gatech.edu

Dr. Ashraf Saad
Georgia Institute of Technology
210 Technology Circle
Savannah, GA 31407
912-966-7916
asaad@gtsav.gatech.edu

ABSTRACT

Society has grown to rely on Internet services, and the number of Internet users increases every day. As more and more users become connected to the network, the window of opportunity for malicious users to do their damage becomes very great and lucrative. The computer industry is combating the rising threat of malicious activity with new hardware and software products such as Intrusion Detection Systems, Intrusion Prevention Systems, and Firewalls. However, malicious users are constantly looking for ways to by-pass the security features of these products, and many times they will succeed. This paper describes a novel concept implemented for the purpose of computer and network security with hopes of using it to combat malicious user activity. A hybrid-intelligent system based on Bayesian Learning Networks and Self-Organizing Maps was created and used for classifying network and host based data collected within a Local Area Network. The KDD-CUP-99 data set was used to test this classification system, and the experimental results show that there is an advantage to using a hybrid system such as this because there was a significant improvement in classification accuracy compared to a non-hybrid Bayesian Learning approach when network-only data is used for classification.

Categories and Subject Descriptors

C.2.0 [**Computer Systems Organization**]: Computer-Communication Networks-*Security and Protection*

General Terms

Security

Keywords

Intrusion Detection Systems, Hybrid-Intelligent Systems, Bayesian Learning, Self-Organizing Map

1. INTRODUCTION

Over the past few years, there has been accelerating growth in the use of network based services. Because of the ubiquity of computer systems and the Internet, this growth will continue. It is becoming increasingly common for many of the products that consumers purchase to be capable of Internet connectivity. These products will range from home appliances such as refrigerators and HVAC systems to automobiles. Corresponding to this continued growth in network usage, there will be increases in malicious user activity. The threat of malicious users is becoming more serious every day. Terms such as phishing, pharming, and identity theft make their way into news headlines all the time. The Internet community is seeing malicious activity enter a new level of sophistication. The tools and concepts of malicious users are entering into the underground crime scenes where hackers are being paid for their services [14]. Corporate espionage, extortion, and identity theft are very lucrative commodities within the Internet underground. Therefore, there exists an engineering need to develop intelligent software and hardware systems that can reliably detect malicious user activity.

In order to obtain a suitable order of reliability, the systems for anomaly detection within networked systems will need a high degree of intelligence. There have been systems developed based on intelligent system concepts such as Bayesian Learning Networks (BLN), Self-Organizing Maps (SOM), Decision Trees (DT), and Artificial Neural Networks (ANN) [1], [2], [8], [9], [15]. Some of these previous works have used the University of California, Irvine's Knowledge Discovery in Databases (KDD) datasets to analyze the use of intelligent system algorithms in the field of computer-network security.

This paper discusses the implementation details and experimental results of using a Hybrid Intelligent System for detecting network anomalies due to malicious user activity. A hybrid intelligent system is developed by incorporating the use of a Self-Organizing Map (SOM) with a Bayesian Learning Network (BLN). For this experiment, the KDD-CUP-99 data set from the 1999 Intrusion Detection contest was used. This has proven to be a very good data set to use for testing machine learning and artificial intelligence (intelligent systems) algorithms designed for intrusion detection systems. The KDD-CUP-99 data set uses a version of the data that was collected by the 1998 DARPA Intrusion Detection Evaluation Program that was conducted by MIT Lincoln Labs. Their objective was to survey and evaluate research in intrusion detection [22]. The captured data was collected on a Local Area Network (LAN) that was a simulation of a US Military LAN under normal use with occasional injections of malicious traffic flows that represented various types of computer and network attacks.

The remainder of this paper consists of a brief overview of BLN and SOM. Then, implementation details of the hybrid intelligent

ACM SE'06, March, 10-12, 2006, Melbourne, Florida, USA
Copyright 2006 ACM 1-59593-315-8/06/0004…$5.00.

system are given followed by the experimental results. It ends with a brief discussion of future work and conclusions that have been drawn.

2. OVERVIEW OF BAYESIAN LEARNING

A Bayesian Learning Network (BLN) is a probabilistic model built on the concept of the Directed Acyclic Graph (DAG) [11]. The DAG is a graph of nodes where each node is a random variable of interest. For example, the variable of interest could be a time delay, packet size, attack type, or a protocol value. Furthermore, the variable could be a discrete type or continuous type random variable. The directed edges on the graph represent dependence relations among the variables. If a directed edge is generated from a node **h** to another node **D**, then **h** is the parent of **D**. The fundamental equation of a BLN is given by Equation 1:

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)} \qquad (1)$$

Equation 1 is known as Bayes theorem. From equation 1, *h* is the hypothesis random variable and *D* is the data random variable. The equation states that the probability of a hypothesis given some data is equal to the probability of the data given the hypothesis multiplied by the probability of the hypothesis and divided by the probability of the data. The *P(h|D)* term is referred to as the “posterior probability”, whereas the other terms are referred to as the “a priori probabilities.” A general BLN structure is known to be NP complete [4]. However, when the nodes of the DAG are independent, the properties of probabilistic independence can be invoked, and a simple BLN known as a Naïve Bayes network can be created. The Naïve Bayes network has one root node (parent node). The root node is referred to as the “unobservable” variable of interest, and its children nodes are called the “observable” variables. Hence, observations can be made, and from the observations, inferences can be drawn.

3. OVERVIEW OF SELF-ORGANIZING MAPS

Self-Organizing Maps (SOM) transform a high dimensional input data domain to elements of a low dimensional array of nodes [6]. The SOM array is a fixed size grid of nodes. The size of the grid is a heuristic that is chosen such that the grid gives the best representation of the input data vector set. Let the input data be represented by a set of real vectors $\mathbf{X = [\ x_1 , \dots , x_r\ ]}$, and let a parametric real set of vectors $\mathbf{M_i = [\ m_{i1} , \dots , m_{ik}\ ]}$ be associated with each element *i* of the SOM grid where both **X** and $\mathbf{M_i} \in \Re^n$. A decoder function, d(**X, M**$_i$), which is defined on the basis of distance between the input vector and the parametric vector is used to define the image of the input vector onto the grid. The decoder function is usually chosen to be either the Manhattan or the Euclidean distance metric. The image is said to be the “winner node” or the Best Matching Unit (BMU) within the SOM grid. The BMU is denoted as the index, c, of the node with a minimum distance from the input vector [6]:

$$c = \arg \min_i \{\ d(\mathbf{X}, \mathbf{M_i})\ \} \qquad (2)$$

The dynamics of the SOM algorithm demand that the $\mathbf{M_i}$ be shifted towards the order of **X** such that a set of values { $\mathbf{M_i}$ } is obtained as the limit of convergence of the following [6]:

$$m_i(t+1) = m_i(t) + \alpha(t) * [x(t) - m_i(t)] * H_{ic} \quad (3)$$

In (3), H_{ic} is a neighborhood function which models “elastic” interconnections between nodes, and it is usually a Gaussian function that decreases with distance from the winner node c. The α(t) term is known as the learning rate of the system. The nature of (3) allows the SOM to cluster and shape itself towards a best description of the input data.

4. IMPLEMENTATION DETAILS OF THE HYBRID INTELLIGENT SYSTEM

The hybrid intelligent system described in this paper was constructed with an SOM and a BLN. The SOM theory was slightly modified for this work. The underlying equations of the SOM theory imply that the SOM is classified as an “unsupervised” training algorithm. Unsupervised training means that the algorithm can work with “un-labeled” data. However, the SOM can be used with labeled data if one desires. For this work, an “attribute” variable was assigned to each node in the SOM grid. This attribute was set as either “normal” for data labeled as normal and “abnormal” for all other data vectors regardless of the associated attack type. When a BMU was found for a given input vector, the label was extracted and the attribute was assigned accordingly. Independence amongst the variables of interest was assumed, and a Naïve Bayesian Learning network was used. There were no mathematical derivations to arrive at the independence conclusion, but other work has shown that the Naïve BLN will perform well for this particular dataset. Figure 1 shows the architecture for the hybrid intelligent system that was developed.

As shown in Figure 1, the system first extracts a subset of the KDD-CUP-99 data. This subset of data was a mixture of traffic flows that contained at least one of each type of the various attacks contained in the dataset. However, about 92% of the data subset was traffic flows that were considered to be normal. This subset was used to train the SOM. The theory behind the SOM dictates that similar nodes will tend to cluster together within the map. Hence, if all one cares about is the ability to classify traffic flows between normal and abnormal, biasing the map to one particular type of flow will suffice. And, this reduces the overhead of trying to classify each and every type of traffic flow. After the SOM has been trained, the remaining set of data is sent through the trained SOM. The trained SOM will make its prediction of the nature of the flow, i.e. is it abnormal or normal. Then, the SOM module will append its prediction to each input record. After the remaining data has been processed and modified by the SOM, the data is sent into the Bayesian structure development module which develops a structure file and a processed data file suitable for the Bayesian trainer module. The Bayesian trainer module takes the structure file and processed data and learns the Conditional Probability Tables (CPT) for the Bayesian network. After this has been done, the SOM and BLN have been trained and are ready to classify the test data. The test data are fed into the Bayesian/SOM classifier one record at a time. First, the SOM makes its prediction and appends its result to the record. Then, the BLN makes a probabilistic calculation as to what type of traffic flow it is. The calculated probability and the predicted traffic flow type are appended to the record and output into a classification file.

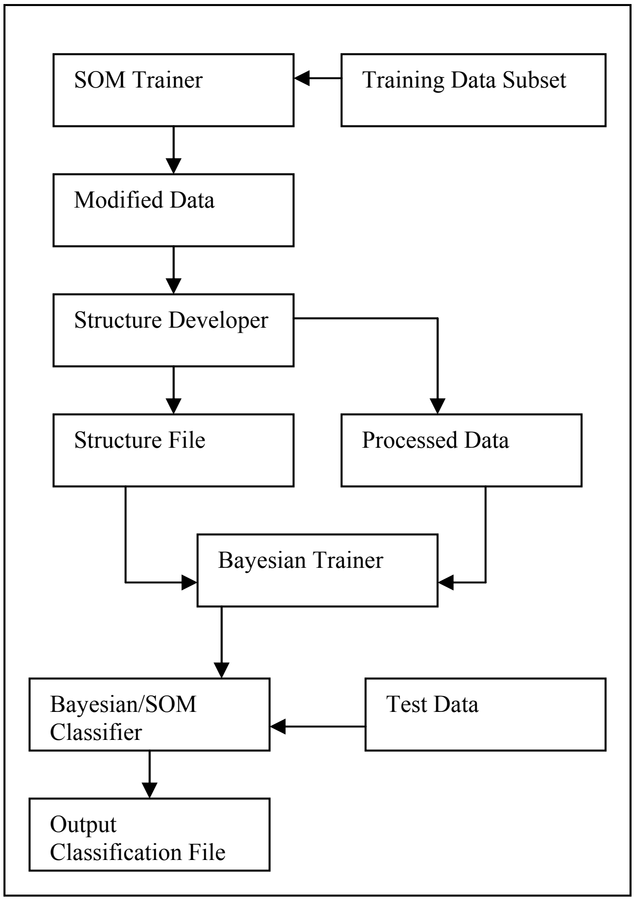

Figure 1. The Hybrid System Architecture

The classification file is the final result of the system. This file can be used for further analysis of the system by an administrator. However, it would be better to use the data in real time. For example, a simple Intrusion Detection System (IDS) could be created by using the calculated probability and predicted traffic flow type. For example, if the classifier predicts that a given flow is not normal with a probability of 99%, then an alert could be sent to a Resource Management System (RMS) that could block the incoming packet stream from the given source Internet Protocol (IP) address. Other types of uses could be considered as well.

5. EXPERIMENTAL RESULTS

For this experiment, 4 types of analyses were made with the dataset. The KDD-CUP-99 data set contained 2 types of variables: 1) host based variables and 2) network based variables. The network based variables were data that could be collected purely within the network, i.e. the TCP protocol number for a given traffic flow. The host based variables were data that could be collected at a given host, i.e. the number of login attempts on a given machine. The 4 analyses constituted the following types: 1) BLN analysis with network and host based data, 2) BLN analysis with network only data, 3) hybrid BLN-SOM analysis with host and network based data, and 4) hybrid BLN-SOM analysis with network only data. The data in Table 1 shows the overall classification accuracies for the 4 types of analyses.

Table 1. Classification Accuracies of the Experiment

	BLN-Host/Network Based	BLN-Network Based	Hybrid-Host/Network Based	Hybrid Network Based
Total Cases	65,505	62,047	65,505	62,047
Correctly Classified	65,019	59,734	65,238	61,631
% Accuracy	99.26%	96.27%	99.59%	99.33%

The results shown in Table 1 reveal that both approaches to classifying the types of traffic flows contained within the data set produce very accurate responses. However, there was a noticeable gain in classification accuracy for the network only based analysis between using a pure BLN versus the Hybrid system. The classification accuracy increased from 96.27% to 99.33%. This was an impressive gain with promising results for one interested in classifying network anomalies using only network based data.

6. FUTURE WORK

The results of this work are very promising. Hybrid intelligent systems can be used for detecting anomalies within a computer network system with a high degree of classification accuracy. However, as with most intelligent system algorithms, the accuracy of the output of these systems depends on how closely the input data matches the data used during the training stage. The question remains as to how well the system will respond to test data that contains a large amount of noise or when the training data becomes "obsolete", i.e. large amounts of data are injected into the classification system that are not represented in the training data. This is a big problem in the area of network security as new attack types are constantly being deployed. This implies that we need a system that can adapt and evolve to changing environments, i.e. changing traffic-data flow types. Work has been conducted as in [13] and [16] where evolutional techniques and algorithms have been applied to training intelligent systems, and these concepts could possibly assist in creating systems that are robust to changes in data flows outside the realm of "normal" intelligent system training. This is one area we will be investigating with future work. Also, we have developed a test-bed of computers and networking gear that will be used as a "honey-net" project [20] [21]. We will be using this test-bed to capture data (both host based and network based) in real time. We will be attempting to classify this data using a hybrid intelligent classification system. The results of the system will be used to implement a Resource Management System (RMS). Such an RMS may use additional intelligence to either block (reject) certain traffic or mark it for handling as "suspect" traffic.

7. CONCLUSION

Modern day computer networks must employ mechanisms for securing the components of the network. Malicious user activities will continue to increase and their tools and procedures will become evermore sophisticated as more and more devices are connected to the network. Because of this, novel techniques for detecting malicious activities and network anomalies will need to be employed. This paper discussed one such novel technique by developing a hybrid intelligent system composed of a Self-Organizing Map and a Bayesian Learning Network. The experimental results show that this type of hybrid system produces a highly accurate response when classifying various types of network traffic flows.

8. REFERENCES

[1] Abouzakhar, N., Gani, A., Manson, G., Abuitbel, M., King, D.: Bayesian Learning Networks Approach to Cybercrime Detection. In proceedings of the 2003 PostGraduate Networking Conference (PGNET 2003), Liverpool, United Kingdom, 2003.

[2] Amor, N., Benferhat, S., Elouedi, Z.: Naïve Bayes vs Decision Trees in Intrusion Detection Systems. In proceedings of the 19th Annual ACM Symposium on Applied Computing, 420-424, 2004

[3] Axelsson, S.: Intrusion detection systems: a survey and taxonomy. Technical report 99-115, March 2000

[4] Cooper, G.: Computational complexity of probabilistic inference using Bayes belief networks. Artificial Intelligence, Vol. 42, 393-405, 1990

[5] Copeland, J., Abler, R., Bernhardt, K.: IP Flow Identification for IP Traffic Carried over Switched Networks, Computer Networks and ISDN Systems, 1998

[6] Kohonen, T.: The Self-Organizing Map. In the Proceedings of the IEEE, Vol. 78, Issue: 9, pp. 1464-1480, 1990.

[7] Kohonen, T., Simula, O., Oja, E.: Engineering Applications of the Self-Organizing Maps. In the Proceedings of the IEEE, Vol. 84, No. 10, pp. 1358-1384, 1996

[8] Lee, S., Heinbuch, D.: Training a neural network based intrusion detector to recognize novel attacks. Information Assurance and Security, pp.40-46, 2000

[9] Lichodzijewski, P., Zincir-Heywood, A., Heywood, M.: Host based intrusion detection using self-organizing maps. In the proceedings of the 2002 IEEE World Congress on Computation Intelligence, 2002

[10] Pearl, J.: Probabilistic Reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, Los Altos, CA, 1988

[11] Pearl, J., Russell, S.: Bayesian Networks. Handbook of Brain Theory and Neural Networks, MIT Press, 2001

[12] Riley, G., Sharif, M., Lee, W.: Simulating Internet Worms, In the proceedings of the 12th IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2004), October, 2004

[13] Saxena, A., Saad, A.: Evolving an Artificial Neural Network Classifier for Condition Monitoring of Rotating Mechanical Systems. International Journal of Applied Soft Computing, Elsevier Publishing, 2005

[14] Skoudis, E.: Counter Hack: A step by step guide to computer attacks and effective defenses, Prentice Hall, Upper Saddle River, NJ, 2002

[15] Valdes, A., Skinner K.: Adaptive Model-based Monitoring for Cyber Attack Detection. In proceedings of Recent Advances in Intrusion Detection (RAID 2000), Toulouse, France, 80-92, 2000

[16] Wiggins, M., Saad, A., Litt, B., Vachtsevanos, G.: Genetic Algorithm-Evolved Bayesian Network Classifier for Medical Applications. In the proceedings of the 10th Online World Conference on Soft Computing (WSC'10), 2005

[17] Ethereal: A Network Protocol Analyzer, http://www.ethereal.com/

[18] Georgia Tech Network Simulator (GtNetS), http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS

[19] Snort: The open source network IDS, http://www.snort.org

[20] The Georgia Tech Honeynet, http://www.ece.gatech.edu/research/labs/nsa/honeynet.shtml

[21] The Honeynet Project, http://www.honeynet.org/index.html

[22] The Third International Knowledge Discovery and Data Mining Tools Competition, http://kdd.ccs.uci.edu/databases/kddcup99/task.html